UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Nicole Schmidt

ANONIMIZAÇÃO DE TRAJETÓRIAS

Florianópolis

2023

Nicole Schmidt

ANONIMIZAÇÃO DE TRAJETÓRIAS

Trabalho de conclusão de curso submetido ao Curso de Ciência da Computação para a obtenção do Grau de Bacharel em Ciência da Computação.
Orientador: Prof. Dr. Jean Everson Martina
Coorientador: Msc. Fernanda Oliveira Gomes

Florianópolis

2023

Catalogação na fonte elaborada pela biblioteca da
Universidade Federal de Santa Catarina

A ficha catalográfica é confeccionada pela Biblioteca Central.

Tamanho: 7cm x 12 cm

Fonte: Times New Roman 9,5

Maiores informações em:
http://www.bu.ufsc.br/design/Catalogacao.html

Nicole Schmidt

ANONIMIZAÇÃO DE TRAJETÓRIAS

Este Trabalho de conclusão de curso foi julgado aprovado para a obtenção do Título de "Bacharel em Ciência da Computação", e aprovado em sua forma final pelo Curso de Ciência da Computação.

Florianópolis, 16 de novembro 2023.

Prof. Dr. Jean Everson Martina
Orientador

Banca Examinadora:

Prof. Thaís Bardini Idalino

Msc. Fernanda Oliveira Gomes
Coorientador

Prof. Dra. Carla Merkle Westphall

Dedico este trabalho à minha família e amigos que estiveram presentes e me auxiliaram em diversos momentos durante essa trajetória.

AGRADECIMENTOS

Agradeço a meu orientador prof. Dr. Jean Everson Martina e co-orientadora Msc. Fernanda Oliveira Gomes que acompanharam, com paciência, todo o processo de elaboração deste trabalho. Também gostaria de agradecer a todos que me apoiaram e me incentivaram durante essa etapa.

RESUMO

No mundo digital existem formas de rastrear diferentes localizações por onde usuários de dispositivos móveis passam, e uma das fontes desse rastreamento são os próprios aplicativos instalados nesses dispositivos. A publicação desses dados pode expor a privacidade de um indivíduo de maneira danosa e sem o seu consentimento. Essa restrição, além de outras relacionadas a proteção de privacidade, faz parte do recém criado conjunto de leis da GDPR (EUROPEIA, 2016) e da LGPD (BRASIL, 2018). Uma das ações a serem tomadas para garantir a privacidade dos usuários é anonimizar os dados antes de sua publicação, garantindo que dessa forma o dado não pertencerá a ninguém. Alguns métodos utilizados para alcançar essa anonimidade são a k-anonimidade, l-diversidade, t-proximidade, supressão, generalização e troca de partes de uma trajetória com outra. Essas abordagens podem ser adaptadas e utilizadas em conjunto, permitindo uma análise sobre o impacto que os diferentes métodos tem na qualidade dos dados e entre si. Para isso, fez-se uma revisão teórica dos algoritmos presentes na literatura e dessa revisão se escolheu três propostas alternativas. Elas foram implementadas junto de testes com aprendizado de máquina visando analisar os resultados obtidos em diferentes categorias: o tempo de anonimização, a qualidade dos dados e a distorção introduzida nos dados.

Palavras-chave: Trajetórias, anonimização, privacidade.

ABSTRACT

In the virtual world there are many ways to track different locations where mobile device users pass through, one of the tracking sources are the installed apps on these devices. Publishing this data can expose the privacy of an individual in a harmful way and without consent. This restriction, besides others related to privacy protection, is part of the GDPR (EUROPEIA, 2016) and LGPD (BRASIL, 2018) laws. One option to ensure user privacy is to anonymize the data before its release, ensuring nobody can be linked to it. Some methods used to achieve anonymity are k-anonymity, l-diversity, t-proximity, supression, generalization, and exchange of segments of a trajectory with another. These approaches can be adapted and used together, allowing an analysis of the impact of different methods on data quality and between each other. In order to achieve it, a theoretical review was carried with the algorithms present in the literature and from this review three proposals where chosen. These were implemented together with machine learning powered tests which aim to analyze the results obtained in different categories: anonymization time, data quality and introduced distortion.

Keywords: Trajectories, anonimization, privacy.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

SUMÁRIO

1 INTRODUÇÃO

No mundo digital existem formas de rastrear diferentes localizações por onde usuários de dispositivos móveis passam, uma das fontes desse rastreamento são os próprios aplicativos instalados nesses dispositivos. Em alguns casos esses dados de localização serão posteriormente publicados de forma que outras entidades possam analisá-los e utilizá-los para diversos fins, tais como: tomadas de decisão, rotas eficientes, recomendações de localização e mineração de padrões de locomoção.

Deve-se tomar cuidado para que com a publicação desses dados não seja possível identificar um usuário unicamente. A identificação de um indivíduo pode expor a sua privacidade de maneira danosa e sem o seu consentimento. Essa restrição, além de outras relacionadas a proteção de privacidade, fazem parte do recém criado conjunto de leis da GDPR (EUROPEIA, 2016) e da LGPD (BRASIL, 2018), regulamentações para garantir que a privacidade de usuários não seja violada e as quais aplicativos e empresas que lidem com dados de usuários devem obedecer. Uma das ações a serem tomadas para garantir a privacidade dos usuários é anonimizar os dados antes de sua publicação, garantindo que dessa forma o dado não será associado a ninguém.

Um método conhecido e utilizado é o da k-Anonimidade que se baseia em deixar um dado indistinguível de outros k-1 dados. Porém, somente o uso desse método não é suficiente para proteger os dados devido à falta de diversidade que seus resultados proporcionam. Dado essa questão, se faz necessário o uso de outros métodos além da k-anonimidade para alcançar um maior nível de proteção. A diferente combinação de métodos e até a limitação do uso de algumas estratégias como, por exemplo, o uso de supressão, podem ser aplicadas para criar um balanceamento entre a privacidade e a qualidade dos dados sendo processados.

Este trabalho implementa alguns métodos de anonimização e aplica-os a bases de dados contendo trajetórias semânticas com o intuito de avaliar o impacto na qualidade dos dados e na privacidade.

1.1 JUSTIFICATIVA

Com o aumento do uso de aplicativos baseados em localização, a quantidade de trajetórias geradas e coletadas diariamente vem aumentando consideravelmente. Esses dados podem ser publicados e utilizados para análise, mas sem sua anonimização apropriada eles podem expor informações sensíveis de usuários ferindo a sua privacidade. Considerando isso, foi realizado um projeto de pesquisa na área de privacidade e anonimização de trajetórias. A pesquisa foi feita em partes, incluindo a aprendizagem de princípios básicos de privacidade e anonimidade, técnicas de mineração de dados utilizadas para a classificação e o agrupamento de informações e criação de um programa baseado em uma das revisões bibliográficas para a anonimização de trajetórias. Como resultado se obteve um programa de anonimização de trajetórias e seus resultados experimentais. Com esse trabalho feito, faltam análises comparativas entre diferentes métodos de anonimização.

1.2 OBJETIVOS

1.2.1 Objetivos Gerais

Este trabalho pretende avaliar o impacto da utilização de métodos de privacidade na qualidade e proteção de dados de trajetórias semânticas, comparando as bases de dados antes e depois com base na distorção de dados e nas análises utilizando aprendizado de máquina.

1.2.2 Objetivos Específicos

Os objetivos específicos deste trabalho são os seguintes:

- Revisão teórica dos algoritmos presentes na literatura;
- Implementação de pelo menos três algoritmos com diferentes abordagens de anonimização;

- Implementação de testes para análise dos resultados dos algoritmos implementados;

- Análise dos resultados obtidos pelos algoritmos.

1.3 LIMITAÇÕES

Em relação às limitações deste trabalho, considera-se a qualidade do conjunto de dados utilizada na avaliação dos algoritmos, o que é melhor elaborado no Capítulo 4. Outra questão foi o tempo de anonimização dos algoritmos e de processamento do nível de risco dos dados anonimizados apresentado no Capítulo 7. Por fim, considera-se que a implementação feita dos algoritmos escolhidos não é idêntica ao original proposto pelos artigos já que no material acessível encontrado há muitas simplificações do código exposto e algumas omissões de como estruturas deveriam ser implementadas.

1.4 METODOLOGIA

Este trabalho foi elaborado a partir de uma pesquisa exploratória com o intuito de trazer uma perspectiva sobre o impacto de diferentes métodos de anonimização da literatura.

Em um primeiro momento, é revisado o estado da arte de anonimização de trajetórias, fazendo também um levantamento dos conceitos básicos de privacidade, anonimização, trajetórias, etc. Em seguida, a partir da revisão do estado da arte foram escolhidos três algoritmos com métodos de anonimização diferentes para implementar.

Na validação dos resultados, a metodologia empregada foi experimental para avaliar o impacto de métodos de anonimização diferentes sobre um mesmo conjunto de dados. Dada a natureza dos resultados envolverem uma análise probabilística tanto para a qualidade quanto para a privacidade dos dados tem-se a utilização de uma abordagem quantitativa nesse aspecto.

Com os algoritmos implementados, compara-se o antes e depois da anonimização de cada algoritmo com uma base de dados. Para essa comparação testa-se a

previsibilidade dos dados com o uso de aprendizado de máquina a fim de entender a distorção desse comportamento depois da anonimização e quais outros impactos podem ter acontecido. Também se aplica um conjunto de ataques nos dados anonimizados para analisar o resultado dos métodos de anonimização.

1.5 ESTADO DA ARTE

Este capítulo apresenta a seleção de trabalhos relacionados através de uma revisão sistemática e do uso de métodos na anonimização de conjuntos de dados de trajetórias, tendo um enfoque nos métodos que anonimizam trajetórias semânticas. No capítulo 3 se apresenta os métodos selecionados de forma resumida constatando o objetivo que o artigo se propunha a resolver, o funcionamento do método e os seus resultados obtidos.

1.5.1 Revisão Sistemática

A revisão sistemática foi realizada acerca dos trabalhos relacionados com a anonimização de trajetórias semânticas. O objetivo é avaliar os diversos métodos propostos por diferentes artigos nas questões de qualidade dos dados resultantes tanto na utilidade dos dados quanto na privacidade dos indivíduos a quem esses dados pertençam. Com isso, elaboraram-se as seguintes perguntas:

- Quais são os métodos recentes utilizados para a anonimização de trajetórias?
- Dentre os métodos utilizados, qual o impacto deles na utilidade dos dados e na privacidade dos usuários após o processo de anonimização?

As palavras-chave pesquisadas foram: *("privacy"OR "k-anonymity"OR "anonymity"OR "anonymization"OR "privacy-preserving"OR "privacy preserving"OR "privacy protection") AND ("trajectory"OR "moving objects"OR "location based"OR "location"OR "trajectories"OR "spatio-temporal"OR "mobility") AND ("semantic"OR "points of interest"OR "POI"OR "diversity")*

A partir da elaboração das palavras-chave esquematizadas na Tabela 1 foram selecionados artigos a partir dos seguintes sites: IEE Xplore Digital Library, ACM Digital Library, Science Direct, Springer Link e Google Scholar.

Palavra-Chave	Sinônimos
privacy	k-anonymity, anonymity, anonymization, privacy-preserving, privacy preserving, privacy protection
trajectory	moving objects, location based, location, trajectories, spatio-temporal, mobility.
semantic	points of interest, POI, diversity.

Tabela 1 – Palavras-chave e sinônimos.

Site	Pesquisa inicial
IEEE	238
ACM	10,222
Springer Link	14,675
Science Direct	7,521
Google Scholar	$\sim$ 1.210.000

Tabela 2 – Resultados da revisão sistemática.

Para selecionar os artigos se considerou o *qualis* dos eventos e periódicos no qual eles foram publicados. Após essa separação, trabalhos com métodos similares ou que trabalhavam somente com a anonimização de dados online foram desconsiderados. Um sumário da revisão literária pode ser visto no Capítulo 3.

1.6 ORGANIZAÇÃO DOS CAPÍTULOS

O trabalho foi organizado em oito capítulos. O primeiro capítulo apresenta o contexto, justificativa, objetivos, limitações, metodologia e estado da arte. O segundo capítulo realiza uma revisão bibliográfica dos conceitos básicos utilizados no

contexto de anonimização de trajetórias. O terceiro capítulo discorre sobre trabalhos correlatos, comentando sobre diversas abordagens utilizadas como: supressão de trajetórias, troca de segmentos parciais de trajetórias, reconstrução de trajetórias, entre outros. O quarto capítulo traz a proposta deste trabalho e qual o conjunto de dados utilizado no seu contexto. O quinto capítulo apresenta os algoritmos escolhidos por meio de uma breve explicação e detalha como foi realizada a sua implementação. No sexto capítulo são apresentados os experimentos realizados em cima dos dados anonimizados. No sétimo capítulo se comenta sobre os resultados obtidos a partir do capítulo anterior. No último capítulo é apresentada a conclusão do trabalho desenvolvido e os trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

2.1 TRAJETÓRIAS

Uma trajetória é uma sequência de pontos espaço-temporais de um objeto em movimento, em que cada ponto possui a informação sobre o posicionamento do objeto e o tempo em que o objeto se encontrava naquele lugar. Frequentemente um ponto pi é representado por (xi, yi, ti), sendo xi e yi as coordenadas geográficas em que o ponto se encontra e ti o tempo em que foi registrado o acesso. Se considera que a conexão entre dois pontos é uma linha reta.

2.2 TRAJETÓRIAS SEMÂNTICAS

Uma trajetória semântica possui em cada ponto além da informação geoespacial e temporal a informação adicional sobre o lugar em que esse registro ocorreu, como, por exemplo, as coordenadas indicarem que o registro foi feito em uma estação de metrô.

2.3 PRIVACIDADE

De acordo com (RUBENFELD, 1989) a privacidade limita a capacidade dos outros de ganhar, disseminar ou usar qualquer informação sobre um indivíduo. Gavison (1990) complementa em (GAVISON, 1980) que a privacidade é uma limitação do acesso dos outros a um indivíduo. Além disso, (REIMAN, 1976) afirma que privacidade é uma prática social: a capacidade de controlar quem possui acesso a informação pessoal que um individuo decide divulgar ou não o permite manter uma variedade de relações em suas vidas. As diferentes formas de se ver privacidade como um direito são: o direito para ser deixado sozinho, acesso limitado a alguém, segredos, controle sobre informação pessoal, personalidade e intimidade (SOLOVE, 2002).

2.4 ANONIMIZAÇÃO

De acordo com (PFITZMANN; HANSEN, 2005) a anonimidade é um estado de não ser identificável em um conjunto de indivíduos para que assim se proteja a informação pessoal. A anonimidade é desejável porque ela garante privacidade. O processo conhecido como anonimização é usado em bases de dados para garantir que mesmo que eles sejam utilizados para posterior análise a privacidade de seus usuários será mantida. Porém, mesmo que algumas bases de dados usem esse processo, ainda é possível de-anonimizar a informação caso duas bases de dados diferentes sejam combinadas, como foi divulgado em (OHM, 2009).

2.5 APRENDIZADO DE MÁQUINA

Segundo (MITCHELL, 1997), um programa aprende se a partir de uma experiência em uma tarefa ele consegue, com essa experiência, melhorar o seu desempenho na próxima execução dessa tarefa. Em uma modelagem de aprendizado de máquina dita "supervisionada", a partir de um conjunto de treinamento se colhem estatísticas, padrões e com isso os pesos dados aos valores de entrada são ajustados. Assim, quando esse algoritmo tentar responder alguma entrada desconhecida a resposta será baseada em uma estimativa do que ele já conhece.

2.6 K-ANONIMIDADE

K-Anonimidade é um termo introduzido por (SAMARATI; SWEENEY, 1998) que define um método utilizado para garantir que dados não possam ser traçados a um indivíduo. Os quasi-identificadores (QI) existentes nos dados facilitam de forma indireta a re-identificação desses indivíduos e podem ser obtidos conectando diversos atributos disponíveis em fontes de dados públicos, a combinação de algumas características frequentemente é única para identificar alguns indivíduos. O objetivo da k-Anonimidade é identificar esses QIs e anonimizá-los.

2.7 *L*-DIVERSIDADE

A k-anonimidade ainda pode vazar informações devido à falta de diversidade nos atributos sensíveis, o método não garante proteção contra ataques baseados em conhecimento prévio. Para evitar isso, a tabela de dados publicada deve também garantir diversidade, significando que em adição à k-anonimidade todas as tuplas que compartilhem os mesmos valores do quasi-identificador devem possuir valores diversos com frequências similares para os seus atributos sensíveis.

Segundo (MACHANAVAJJHALA et al., 2007), dado um adversário com conhecimento prévio, uma tabela pode vazar informações privadas de dois jeitos: divulgação positiva e divulgação negativa. A positiva acontece quando o adversário pode identificar corretamente o valor de um atributo sensível com uma alta probabilidade, já a negativa acontece quando o adversário pode eliminar corretamente alguns valores possíveis de um atributo sensível. A tabela publicada deve prover para o adversário a menor quantia de informação adicional possível, não havendo uma grande diferença entre suas crenças antes e após analisar a tabela.

Com dados o suficiente e uma boa partição o conhecimento prévio se cancela e não possui nenhum efeito nas inferências que podem ser feitas a partir da tabela. A única inferência que pode ser feita são as que dependem completamente na frequência em que cada atributo sensível aparece em cada bloco. Para prevenir que aconteça quebras de segurança é necessário garantir que para cada bloco os valores mais frequentes de um bloco possuam aproximadamente a mesma frequência, garantindo assim que o adversário terá uma incerteza sobre o valor do atributo sensível que ele deseja descobrir. Essa é a essência da l-diversidade.

2.8 *T*-PROXIMIDADE

A divulgação de informação pode ocorrer tanto com a divulgação de identidade, que é quando um indivíduo é ligado a um registro em particular em uma tabela publicada, quanto a de atributo. A divulgação de atributo acontece quando uma nova informação sobre indivíduos é publicada e esses novos dados possibilitam inferir características sobre um indivíduo com mais precisão, e pode ocorrer sem que ocorra uma divulgação de identidade. A divulgação de um atributo falso também

pode ser danosa quando o observador percebe de forma incorreta o valor de um atributo sensível de um indivíduo e age de acordo com essa percepção prejudicando o indivíduo.

De acordo com (LI; LI; VENKATASUBRAMANIAN, 2007) é possível que um adversário ganhe informação sobre os atributos sensíveis contanto que ele tenha informação sobre a disposição global desse atributo. A t-proximidade formaliza a ideia de um conhecimento prévio geral exigindo que a distribuição de um atributo sensível em qualquer classe equivalente seja próximo da distribuição geral do atributo na tabela. Isso limita a quantidade de informação específica do indivíduo que um observador pode aprender. Se uma classe equivalente tem um valor que aparece muito mais frequentemente que qualquer outro isso irá permitir que o adversário conclua que uma entidade na classe de equivalência tem uma maior chance de ter aquele valor.

É possível que ocorra o vazamento de informações sensíveis na tentativa de garantir a diversidade dos valores sensíveis em cada grupo quando não se considera a proximidade semântica desses valores. Isso torna o dado suscetível a um ataque de similaridade pois o valor de um atributo sensível em uma classe de equivalência é distinto, mas os valores na classe são semanticamente similares, permitindo assim que um adversário consiga aprender uma informação importante. Distribuições com o mesmo nível de diversidade podem prover diferentes níveis de privacidade devido à relação semântica entre os valores dos atributos. Valores diferentes possuem diferentes níveis de sensibilidade e a privacidade também é afetada pelo relacionamento entre os valores dos atributos.

Exigir que dois atributos sejam próximos limitaria a quantidade de informação útil que é divulgada pois isso diminui a correlação entre os atributos quasi-identificadores e os atributos sensíveis. Contudo, se um observador obtém uma imagem muito clara dessa correlação, então a divulgação do atributo ocorre. O parâmetro t em t-proximidade permite que ocorra uma troca entre a utilidade e a privacidade.

t-proximidade protege contra a divulgação de atributo mas não é boa com a divulgação de identidade, nesse sentido é desejável usar tanto t-proximidade quanto k-anonimidade. t-proximidade lida com o ataque de conhecimento prévio ao garantir que se um ataque ocorrer, então ataques similares podem ocorrer mesmo com uma tabela totalmente generalizada.

2.9 POIS

Points of Interest (POIs) são pontos da trajetória de um usuário que revelam o valor semântico do lugar e na qual pode-se obter alguma informação do usuário a partir de quanto tempo ele permaneceu nesse determinado ponto de interesse (ZHANG et al., 2015). É possível tratar a informação desses pontos para que ela se torne mais genérica, então ao invés de aparecer que o usuário estava em um restaurante generaliza-se essa informação para dizer que ele estava em um local de comércio.

2.10 ROIS

Regions of Interest (ROIs) são as regiões formadas a partir dos POIs e pode ser representada pelo seu ponto central, raio e tempo de duração (ZHANG et al., 2015). No geral os ROIs ajudam a garantir que aquela região possui uma boa distribuição de POIs.

2.11 SIMILARIDADE

A similaridade entre duas trajetórias é muito considerada quando os algoritmos precisam verificar qual é o custo de unir duas trajetórias para satisfazer a k-anonimidade, quanto de informação vai ser perdida pela generalização de alguns aspectos da trajetória. Esse cálculo pode ocorrer considerando a similaridade ponto a ponto da trajetória (TU et al., 2017) ou a partir da quantidade de ROIs que as duas trajetórias tem em comum (ZHANG et al., 2015).

2.12 CLASSIFICAÇÃO

A classificação é uma técnica que emprega um algoritmo de aprendizagem para identificar um modelo que combine melhor com a relação entre o conjunto de atributos e o rótulo de uma classe dos dados de entrada. Os algoritmos possuem

duas etapas: treinamento e teste. A etapa de treinamento consiste em separar uma porção dos dados para gerar o classificador enquanto o treinamento será utilizado para avaliar o quão efetivo esse classificador gerado é. Quanto maior for o conjunto de treinamento melhor será o classificador, e quanto maior o conjunto de teste mais precisa será a estimativa do desvio padrão de erro.

Existem dois tipos de classificação em mineração de dados: os de aprendizagem ansiosa e os de aprendizagem preguiçosa. Métodos de aprendizagem ansiosa começam a aprender quando eles recebem o conjunto de dados, não esperando pelo conjunto de teste para aprender. Esses métodos demoram um longo tempo aprendendo e pouco tempo classificando. Árvores de decisão, *Naive Bayes*, máquina de vetores de suporte e redes neurais artificiais são exemplos de aprendizagem ansiosa. Os métodos de aprendizagem preguiçosa armazenam o conjunto de dados sem aprender dele, apenas classificando quando recebem o conjunto de testes. Esses métodos gastam pouco tempo treinando e mais tempo classificando dados. Alguns exemplos são *K-nearest neighbor* e *Case-based reasoning*.

Se utilizam algumas métricas para julgar o desempenho do modelo, entre elas temos a precisão, erro, sensibilidade, cobertura, revocação e medida F. A precisão e o erro utilizam de base uma matriz de confusão exemplificada na Figura 1, a precisão é dada pela soma dos positivos e dos negativos verdadeiros dividido pelo número total de resultados e o erro é calculado de uma forma similar, considerando os falsos positivos e negativos no lugar dos verdadeiros.

2.12.1 Árvores de Decisão

As árvores de decisão são representações gráficas que consistem de um nó raiz, ramos e nós-folha. O nó raiz representa um atributo dos dados de entrada, cada nó-folha representa uma classe, os nós internos representam testes de atributo, e os ramos são os resultados desses testes. Depois do nó raiz, o próximo atributo para teste é escolhido a partir da entropia (ganho de informação), que é a medida de aleatoriedade na informação sendo processada, do índice de Gini ou do erro de classificação (TAN et al., 2018). No caso da entropia, quanto maior ela for, mais difícil é de tirar qualquer conclusão do resultado de um atributo. Os nós-folha recebem o nome da classe caso todos os registros de treinamento sejam da mesma classe.

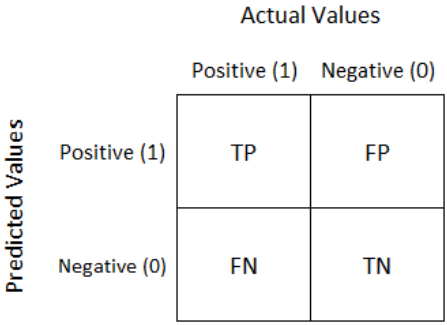


Figura 1 – Matriz de confusão. (NARKHEDE, s.d.)

2.12.2 Naive Bayes

Naive Bayes é uma técnica de classificação baseada no Teorema de Bayes com uma suposição de independência entre seus preditores e providencia uma forma de calcular a probabilidade posterior (WIKIPÉDIA, s.d.[b]). A técnica assume que a presença de uma característica particular em uma classe não é relacionada à presença de qualquer outra característica. O modelo é particularmente útil para um grande conjunto de dados.

Ao classificar um objeto a probabilidade dele pertencer a uma certa classe é o número de objetos daquela classe dividido pelo número total de objetos. A classificação final é feita por se considerar a informação de todas as probabilidade de se parecerem com uma certa classe usando a condição de probabilidade do teorema de Bayes.

O algoritmo começa convertendo o conjunto de dados em uma tabela de frequências, então se cria uma tabela de probabilidade ao se encontrar as probabilidades. Por último ele usa a equação Naive Bayesiana para calcular a probabilidade posterior de cada classe, tomando a que tiver a maior probabilidade posterior como resultado da previsão. O método possui uma boa performance em predições de várias classes e quando se tem a suposição de uma independência não se precisa de tanto

treinamento. As desvantagens consistem no caso de uma variável categórica possuir uma categoria que não foi observada no conjunto de treinamento pois o modelo irá lhe designar uma probabilidade zero e não será capaz de fazer uma predição. Esses casos são conhecidos como "Frequência Zero". O método também possui limitações pois supõe que os preditores são independentes, o que é improvável de acontecer na vida real.

2.12.3 Redes Neurais Artificiais

As redes neurais são usadas para transformar dados não processados em informação útil, procurando por padrões em grande quantidades de dados. As redes Neurais Artificiais (ANN) também são conhecidas como ferramentas de modelagem de dados estatísticos não-lineares que reconhecem padrões e respondem apropriadamente. ANN são boas observadoras devido ao seu gerador de peso sináptico, resultando em uma alta precisão e performance. O método consiste de três partes: a sua arquitetura, o algoritmo de aprendizagem e a função de ativação. A arquitetura é como os neurônios são organizados e como eles interagem um com o outro. No algoritmo de aprendizagem, ocorre o treinamento da ANN para realizar tarefas especificas. Por último, a função de ativação, também conhecida como transferência, é executada em cima do resultado gerado pelo nó (WIKIPÉDIA, s.d.[a]).

O reconhecimento de padrões é dependente ou do que está acontecendo ou de qual é o histórico dos dados. As redes neurais artificiais também podem ser utilizadas em processos de classificação e agrupamento na mineração de dados, o seu classificador recebe de entrada os atributos de classificação. Todos os dados devem ser normalizados, os valores mudam para pertencer ao intervalo de [0,1] ou de [-1,1]. As entradas possuem pesos nelas para ajudar com a estimativa. Se o resultado for o esperado então o algoritmo vai manter os pesos atuais, caso contrário ele irá os reajustar e computar novamente outro resultado.

2.12.4 Máquina de Vetores de Suporte

A Máquina de Vetores de Suporte (SVM) é um método matemático baseado no teorema de Lagrange utilizado para a classificação e a regressão de tarefas e

se origina da teoria de aprendizagem estatística. O método gera partições sem sobreposição e usualmente emprega todos os atributos. O espaço da entidade é particionado em um passo único, gerando partições planas e lineares. O método é baseado na margem linear máxima de discriminantes, mas ele não considera a dependência entre atributos. Sua desvantagem é a sensibilidade ao ruído e a demora para gerar um modelo e executá-lo. Além disso, os parâmetros influenciam muito nos resultados (AKASHKUMAR17, s.d.).

O classificador é formalmente definido por um hiperplano de separação. A partir de dados de treinamento rotulados o algoritmo irá retornar um hiperplano ideal que categoriza novos exemplos. O hiperplano selecionado deve ser aquele que divide melhor as classes. O plano que possuir a maior margem (distância entre o ponto de dado mais próximo do hiperplano) será considerado o correto para a classificação das classes.

2.12.5 Nearest Neighbour

O K-Nearest Neighbour é um algoritmo que armazena todos os possíveis casos e classifica eles em um novo baseado na medida de similaridade. Para escolher o valor ideal de K é necessário inspecionar os dados primeiro. Em geral, um valor bastante grande é mais preciso já que reduz em geral o ruído. A validação cruzada é outra maneira de determinar um bom valor de K ao usar um conjunto de dados independente para validá-lo. O valor costuma ficar entre três e dez.

É um método simples que consiste na procura de K registros próximos ao que está sendo classificado. O algoritmo, em vez de comparar dados não classificados com todos os outros, executa um cálculo matemático para medir a distância entre os dados e fazer a classificação. O método começa recebendo dados não classificados e medindo a distância entre o dado novo e todos os outros que já foram classificados. Em seguida, ele parte das K menores distâncias e verifica as classes que tiveram a menor valor distância e conta a quantas vezes cada classe aparece. A classe que aparecer mais vezes será escolhida como a correta (NAVIANI, s.d.).

2.13 TIPOS DE ATAQUE

Os ataques utilizados para testar os dados foram baseados em (PELLUNGRINI et al., 2017) e a implementação utilizada foi fornecida pela biblioteca (PAPPALARDO et al., 2022).

2.13.1 Ataque de Lugar

Um ataque de lugar é um cenário no qual o atacante conhece uma certa quantidade de lugares visitados por um indivíduo sem saber a ordem na qual o indivíduo visitou esses lugares.

2.13.2 Ataque de Sequência de Lugar

Em um ataque de sequência de lugar o atacante conhece um subconjunto de localizações visitadas pelo indivíduo e a ordem temporal das visitas.

2.13.3 Ataque de Tempo de Lugar

Ataque de tempo de lugar, ou ataque de visita, ocorre quando o atacante conhece um subconjunto de lugares visitados por um indivíduo e o horário que o indivíduo esteve nesses lugares.

2.13.4 Ataque de Lugar Único

Nesta situação o atacante conhece as coordenadas de um lugar único visitado por um indivíduo.

2.13.5 Ataque de Frequência

Neste caso o atacante conhece os lugares visitados pelo indivíduo, a sua ordem recíproca de frequência e o número mínimo de visitas que o indivíduo fez a esses lugares.

2.13.6 Ataque de Probabilidade

Em um ataque de probabilidade o atacante conhece os lugares visitados por um indivíduo e a probabilidade desse indivíduo de visitar cada lugar.

2.13.7 Ataque de Proporção

Um ataque de proporção assume que o atacante conhece um subconjunto de lugares visitados por um indivíduo e também a proporção relativa entre o número de visitas nesses lugares. O atacante sabe a proporção entre a frequência do lugar conhecido mais visitado e os outros lugares conhecidos.

2.13.8 Ataque de Casa e Trabalho

Este ataque considera que o atacante conhece os dois lugares mais frequentes de um indivíduo e as suas frequências.

3 TRABALHOS CORRELATOS

3.1 ESTUDO DE ALGORITMOS DE ANONIMIZAÇÃO DE TRAJETÓRIAS

Para a elaboração deste trabalho foram analisados artigos para contemplar algumas técnicas utilizadas na anonimização de trajetórias, bem como para a revisão teórica. (ABUL; BONCHI; NANNI, 2008) tem como principal contribuição a introdução do conceito de anonimidade (k,δ) e o desenvolvimento de uma medida baseada no agrupamento de trajetórias em *clusters* que minimiza a distorção de informação introduzida pela tradução espacial. A anonimidade (k,δ) considera no parâmetro δ a incerteza da localização de um ponto reportado pelo usuário. Esse parâmetro ajuda a definir a área de um conjunto de trajetórias anonimizado, pois ela é formada pelo menor círculo que engloba todos os pontos do conjunto. O algoritmo começa na formação de clusters escolhendo um pivô e seus k-1 vizinhos não visitados mais próximos, o próximo pivô escolhido é o situado mais longe possível do atual. Caso um pivô não consiga formar um agrupamento um novo pivô será procurado, se alguma trajetória não conseguir fazer parte de um agrupamento ela será descartada. Em seguida, a última etapa é a transformação de cada agrupamento em um conjunto (k,δ)-anonimizado. O algoritmo apresentado se mostrou simples e eficiente, anonimizando 350Mb em alguns minutos em uma implementação em C. No entanto, ele não considera diversidade e introduz distorção significativa nas trajetórias anonimizadas. Os autores mencionam também o problema do parâmetro δ ser considerado uniforme e propõem algumas alterações, mas não levantam resultados sobre. Em (HUO et al., 2012) é proposto proteger a privacidade através de pontos de parada significantes. A abordagem usa métodos de generalização, agrupamento e de pontos da trajetória que não são cobertos por zonas. O algoritmo considera trajetórias semânticas e utiliza do conceito de l-diversidade. Antes de anonimizar as trajetórias se extrai os pontos de parada de cada trajetória com o objetivo de reconstruir os lugares semânticos encontrados a partir desses pontos, com essa reconstrução se cria zonas que contenham no mínimo l lugares. Na anonimização as trajetórias são divididas em sequências de movimento e de parada, os pontos de parada são substituídos pelas zonas correspondentes a sua localização. Caso um ponto de passagem se encontre em uma zona ele será deletado. Há duas abordagens apre-

sentadas para a construção de zonas: o GridPartition e o DiverseClus. Na primeira o mapa é dividido em uma grade de quadrados uniformes, como o número de lugares em cada quadrado da grade é diferente os autores planejaram uma estratégia para garantir que cada quadrado tenha no mínimo l lugares. Quando o quadrado tem essa quantia miníma de lugares ele pode ser considerado uma zona. O DiverseClus tem como objetivo agrupar l lugares em uma zona de área minimizada considerando a dissimilaridade de cada lugar adicionado com os já existentes na zona. O agrupamento realizado por esse método deve passar por um pós-processamento já que algumas zonas resultantes podem acabar se sobrepondo espacialmente. Ambas abordagens de criação de zonas geram menos de 20% de perda de informação, mas o DiverseClus gera menos perda que o GridPartition devido à influência do tamanho das zonas nesse aspecto. A perda de informação em ambos algoritmos de criação de zonas ocorre principalmente pela generalização de pontos de parada e da deleção de pontos de passagem por zonas. Como o artigo tem uma restrição no uso da API do Google Maps a extração de pontos e a geração de lugares leva 2 segundos por coordenada, resultando em uma demora total de 2951 minutos. Apesar da menor perda de informação, a garantia de privacidade do algoritmo proposto nem sempre é melhor que a obtida em abordagens que utilizam da k-anonimidade.

Com relação à supressão de trajetórias, (BRITO et al., 2015) além de utilizar supressão também se baseia na propriedade k^m. A k^m-anonimidade considera que um atacante pode conhecer até m lugares de qualquer trajetória, dando a garantia que para cada combinação de tamanho m no conjunto de dados, um indivíduo é indistinguível de pelo menos $k-1$ outros. O algoritmo proposto Tranon tem duas etapas principais: a descoberta de quasi-identificadores (QIDs) através do paradigma MapReduce e a anonimização. No MapReduce os dados estão distribuídos em um agrupamento de máquinas e as funções de mapear e de reduzir são aplicadas neles. O mapeamento procura gerar, para cada trajetória, todas as combinações de tamanho i, emitindo um conjunto de pares $< chave, valor >$. Essa informação é utilizada para saber quantas vezes cada combinação aparece no conjunto de dados. A função de redução pega esse mapeamento e agrega em uma soma todos os pares que possuam a mesma chave, emitindo um conjunto de $< chave2, valor2 >$ que consiste em subconjuntos de tamanho i e o número de vezes que esse subconjunto ocorre. Os pares emitidos possuem valores menores que k. Na segunda etapa se escolhe pontos-chave para serem removidos desses conjuntos recém-calculados. Também se seleciona alguns QIDs para serem removidos para diminuir a perda de informação, a

escolha se baseia nos pontos mais infrequentes entre os infrequentes. Os pontos selecionados são considerados o conjunto de acerto do conjunto de quasi-identificadores, dessa forma é dado que o tamanho desse conjunto é equivalente à perda de informação. Para minimizar essa perda se utiliza uma abordagem gulosa para calcular um bom acerto, o conjunto de acerto é selecionado por um algoritmo guloso para ser removido do conjunto de dados original, o cálculo de QIDs de tamanho $i+1$ não leva em conta os pontos removidos em etapas anteriores, o novo ponto mais frequente é recalculado após cada remoção e ele é suprimido do conjunto de dados anonimizados. O algoritmo finaliza a execução quando o conjunto de quasi-identificadores está vazio, dessa forma um ponto é removido de cada QID. O algoritmo proposto anonimiza menos lugares devido ao uso de uma estratégia gulosa para suprimir apenas as localizações-chave. Também se observa a escalabilidade do algoritmo, os dados sintéticos que possuem 130k trajetórias chega a ser 2,7 vezes mais rápido que o algoritmo com a qual os autores comparam, aumentando a quantidade de trajetórias para 400k o algoritmo chega a ficar 95 vezes mais rápido.

Quanto a troca de segmentos de trajetórias, (SALAS; MEGIAS; TORRA, 2018) descreve o algoritmo SwapMob que busca preservar a anonimidade desassociando os indivíduos de suas trajetórias. Dois usuários trocam segmentos de suas trajetórias entre si se eles estiverem co-localizados, para isso os limiares de proximidade e de tempo precisam ser satisfeitos. O conjunto de usuários co-localizados é calculado a partir de um intervalo de tempo, entre as amostras recolhidas se utiliza de uma escolha aleatória para gerar pares de trajetória que realizarão a troca das trajetórias parciais. Os pares são gerados a partir de permutações aleatórias de uma primeira metade do conjunto com a segunda metade restante. Caso a quantidade de números do conjunto seja impar um dos números é deixado fora da geração de pares. A arquitetura é pensada como uma comunicação entre um sistema terceirizado confiável e um provedor de serviço, o sistema de confiança seria o algoritmo proposto SwapMob. O algoritmo funciona da seguinte forma: o usuário envia seus dados cifrados com a chave pública do provedor para o SwapMob que armazena em um vetor os dados essenciais para a anonimização. O vetor é enviado para o provedor que descriptografa os dados e adiciona a um novo vetor a localização, armazenando também todos os carimbos de tempo de determinados intervalos das trajetórias. Após isso, o provedor envia um conjunto de trajetórias que estiverem no limiar de distância e tempo definidos previamente. O SwapMob calcula quais trocas devem ser feitas mantendo o ID trocado e o pseudônimo correspondente. Após o período

pré-determinado o sistema de confiança envia a lista de registros e pseudônimos para o servidor. A anonimização do algoritmo se dá através da desassociação de segmentos de trajetórias do usuário que as gerou, mantendo a quantidade de trajetórias que passa por um local o mesmo, mas alterando o caminho original das trajetórias. Após o processo de anonimização as localizações inferidas divergem das originais e não se consegue mais identificar os seus locais de origem nem re-identificar uma trajetória a partir de segmentos conhecidos. É possível que essa proteção seja revertida caso o provedor de serviço compartilhe informações com o algoritmo SwapMob no intuito de garantir a responsabilidade de ambos os participantes. Trajetórias que não cruzam com ninguém em seu caminho não são anonimizadas. O intuito desse artigo é voltado para a geração de mapas de mobilidade e predições que podem ajudar em sistemas de transporte inteligente. Como é notado pelos autores, a alteração no caminho original das trajetórias causa uma perda na utilidade de mineração de trajetória individual.

(NAGHIZADE et al., 2020) também utiliza trocas. As trajetórias são divididas em pontos de movimento e de parada, para cada trajetória se extrai um conjunto de pontos de parada. Cada ponto possui um nível de privacidade próprio que, como o algoritmo se propõe a levar em conta o nível de privacidade exigido pelo usuário, pode ou não ser suficiente. Caso o ponto não possua uma privacidade aceitável se procura algum POI próximo para realizar a alteração na trajetória. Essa fase de trocas de POI considera a semântica dos pontos de parada já existentes na trajetória de forma que a trajetória não indique que, por exemplo, o usuário parou em três restaurantes em um percurso de duas horas. Nessa troca se procura primeiramente os POI com um menor nível de privacidade, posteriormente se filtra considerando a variedade de POIs presentes na trajetória. O objetivo é manter as características gerais da trajetória: a sua duração, proximidade espacial e velocidade. Há duas abordagens para que a troca de pontos de parada aconteça: a troca exaustiva e a Flip Flop. A troca exaustiva é quando ocorre a substituição de um ponto de parada por um POI existente dentro da trajetória, sem alterar seu trajeto original. Se procura por todos os POIs menos sensíveis que o atual na trajetória, um exemplo seria trocar a duração de movimento antes e depois do ponto de parada, assim o tempo de deslocamento se manteria o mesmo mas o ponto de parada seria deslocado para outro menos sensível. Essa abordagem precisa garantir que o novo ponto de parada seja um POI. A abordagem Flip Flop ocorre quando não se encontra nenhum POI nos segmentos da trajetória e assim o algoritmo precisa procurar por POI candidatos

próximos da rota original. A região de busca é limitada por uma elipse considerando quaisquer dois pontos de parada como os seus focos. Após a coleta os POIs encontrados são ordenados pelo seu nível de privacidade, os que possuírem menos privacidade são escolhidos e permutados para encontrar quais mudariam menos o caminho original da trajetória, procurando o menor caminho para ir até o novo POI e dele para o próximo ponto de parada da trajetória. Caso não se encontre POIs o suficiente para alterar as paradas originais a área de busca da elipse é aumentada. No caso de haver diversos candidatos para substituírem a parada sensível uma versão que prioriza a privacidade (PFF) procura pelos POIs menos sensíveis e uma versão que prioriza a utilidade dos dados (UFF) seleciona um POI que minimiza a distância final comparada com a trajetória original durante a troca de cada POI. Em regiões com baixa densidade de POIs se prioriza o uso do UFF. A abordagem Flip Flop apresentada possui um aumento de complexidade temporal quanto maior for a restrição de privacidade e isso pode ser utilizado por um adversário para obter informações sobre o nível de privacidade de um usuário. Também se mostra possível identificar mudanças em sua configuração de privacidade, podendo expor a privacidade de suas viagens. É notável que a necessidade de procurar um POI fora da rota original causa uma mudança nas propriedades temporais e espaciais da trajetória. Como o artigo traz, apesar da mudança acontecer os resultados apresentam uma alta usabilidade dos dados. Os autores notam que, devido ao impacto da densidade de POIs nas buscas UFF e PFF, é possível utilizar a densidade em uma viagem para indicar aos indivíduos uma estimativa do quão privada essa viagem pode ser, permitindo prever o nível de privacidade alcançável antes de divulgar a viagem.

Com respeito a utilização de regiões para garantir a privacidade, (GRAMAGLIA; FIORE, 2015) propõe o algoritmo GLOVE que visa anonimizar o movimento de microdados extraídos de tráfego móvel. De forma inicial, o algoritmo calcula, para cada trajetória, o esforço necessário para juntar uma trajetória com todas as outras e armazena o resultado em uma matriz. A partir da matriz se escolhe um par de trajetórias que possuam o menor valor de junção, que são removidas tanto da matriz quanto do conjunto de dados e juntadas em uma única trajetória com um contador de quantas trajetórias a compõe. Ela é então incluída no conjunto de dados e tem o esforço de junção calculado para cada trajetória na matriz. Esse processo se repete até que todas as trajetórias estejam k-anonimizadas. A junção de trajetórias escolhe a mais longa e procura o ponto de menor custo de junção na trajetória mais curta. Se algum ponto da trajetória menor não for escolhido ele é

juntado com os pontos que foram na etapa anterior. A junção expande o ponto para que ele cubra o menor e o maior valor dos intervalos temporais e geográficos. A trajetória anonimizada resultante passa por um processo de remodelagem que lida com as sobreposições temporais, sejam elas parciais ou completas, criando um ponto para cada caso. A região é obtida juntando a área sobreposta dos pontos que ela substitui. A versão do algoritmo não otimizada consegue anonimizar um conjunto de dados com $k = 2$ em aproximadamente 60 horas em uma máquina de baixo custo e tem que seus cálculos são altamente paralelizáveis. Apesar de $k = 2$ garantir uma indistinguibilidade, os resultados demonstram que a acurácia da posição e a porcentagem de pontos com uma acurácia melhor que dois quilômetros decai com o aumento de k, sendo que os dados obtidos com $k > 5$ ficam dificilmente usáveis. Analisando o uso da técnica de supressão, descartar 8% dos pontos gera uma melhora na acurácia espacial de 5 quilômetros para 1 quilometro e descartando apenas 4% dos pontos a acurácia temporal é reduzida pela metade. Isso reforça bastante como a técnica de supressão auxilia na diminuição da distorção espacial ao eliminar pontos distantes e externos. Percebe-se também que conjunto de dados menores tem uma acurácia geral maior após a anonimização, pois as trajetórias possuem no geral uma duração de tempo menor o que as torna mais fáceis de juntá-las e conjuntos com poucos usuários tendem a ser mais complexos de anonimizar pois a chance de se ter k trajetórias similares diminui.

(ZHANG et al., 2015) baseia-se em ROIs para a anonimização de trajetórias. A primeira etapa para se construir essas regiões é a extração de POIs com base no conjunto de dados. POIs consistem nos pontos iniciais e finais da trajetória, pontos de parada e pontos de virada. Pontos de virada ocorrem quando o ângulo de dois segmentos adjacentes ultrapassa um limiar definido previamente. A intenção é reconhecer e proteger esses pontos sensíveis. O processo de extração identifica e adiciona os pontos ao conjunto de POIs. A próxima etapa agrupa POIs que estão próximos temporal e espacialmente. Para isso se procura um ponto com uma vizinhança para se construir o agrupamento, se o número de POIs na vizinhança for menor que o limiar de densidade o ponto encontrado é considerado um possível candidato a ruído. Caso contrário, ele é um ponto central e a partir dele será construído um novo agrupamento. Após todos os pontos serem visitados se remove todos os candidatos a ruído que não pertençam a um agrupamento. Para evitar distorção nos dados procura-se adicionar o máximo possível desses pontos em agrupamentos já formados. O resultado de cada agrupamento de POIs resulta em um ROI, sendo

o ponto central o centro desse agrupamento. O raio da região e a diferença de tempo é a diferença máxima entre o ponto central e os outros pontos do agrupamento. O conjunto de trajetórias k-anonimizadas é encontrado com a similaridade entre duas trajetórias, dada pelo número de ROIs em comum de duas trajetórias com o número total de ROIs pelos quais essas trajetórias passam. Trajetórias que não pertencem a nenhum dos grupos devem ser adicionadas a grupos existentes o mais longe possível, se a similaridade dessa trajetória com qualquer trajetória pertencente a um grupo for mais de α então ela pode ser adicionada ao agrupamento. Se algum grupo de trajetórias tiver um tamanho menor que k o grupo será removido do conjunto de dados anonimizados. O algoritmo aplica a troca de POIs dentro de cada ROI, em que POIs são trocados aleatoriamente entre si até que todos tenham sido alterados pelo menos uma vez. Depois da troca as trajetórias reconstruídas são adicionadas por fim ao conjunto anonimizado. Os experimentos mostram uma menor distorção espacial devido à troca de POIs ocorrer somente em uma mesma região. Tem-se também que menos pontos são descartados já que a distância das trajetórias não é considerada no agrupamento, se elas passarem pelas mesmas regiões elas serão agrupadas juntas, resultando em uma menor distorção do resultado das consultas aos dados comparado com o resultado anterior à anonimização.

(TU et al., 2017) também usa POIs generalizados. As técnicas de generalização espaço-temporal e de supressão são utilizadas para seguir os critérios da k-anonimidade, da l-diversidade e da t-proximidade. O algoritmo considera uma matriz de custo de unificação que representa o impacto da junção de duas trajetórias, calculado entre cada ponto para todas as trajetórias presentes no conjunto de dados. Nessa etapa se preocupa somente em satisfazer o critério da k-anonimidade. A cada iteração se encontra um par de trajetórias que possuam o menor custo possível na matriz e se junta essas trajetórias em uma nova e anonimizada. As trajetórias recém anonimizadas são removidas da matriz de custo. Caso a trajetória resultante do processo de anonimização não satisfaça o critério da k-anonimidade ela é adicionada à matriz de custo. Se o critério for satisfeito ela é adicionada ao conjunto de dados anonimizados. Esse processo se repete até que todas as trajetórias sejam anonimizadas. A junção das trajetórias ocorre em um processo ponto-a-ponto, se une o primeiro ponto de uma trajetória com o ponto da outra trajetória que possui o menor custo de junção. Pontos que não forem unidos serão juntados com outros pontos já existentes. Após selecionar dois pontos para serem juntados se procura a menor região que os conecta. Se os pontos pertencerem a uma mesma região então

ela será a região resultante, se as regiões forem vizinhas a nova região irá incluir as duas regiões, caso elas estejam a uma distância entre si é necessário adicionar a menor quantia de regiões possível para conectá-las, para o qual foi utilizado o algoritmo de Dijkstra. Para garantir que essa região encontrada satisfaça l-diversidade verifica se ela possui l ou mais categorias de POI, se esse critério não for satisfeito será adicionada mais bases à região até que ela satisfaça a condição. De forma similar, para satisfazer a t-proximidade, se a região não possuir uma distribuição de POI similar com todo o conjunto de dados é necessário adicionar mais bases que ajudem a diversificar a região. Durante a junção de pontos se considera três diferentes casos relacionados com o tempo de início e o tempo em que o ponto permaneceu no mesmo lugar: os tempos não se sobrepõem, eles se sobrepõem em parte ou totalmente. No geral se considera o tempo de início como o menor tempo entre os dois pontos e a duração o maior valor entre os pontos dado o tempo de início somado a sua duração. No fim da junção de pontos ocorre um processo de remodelação para evitar que ocorra uma sobreposição do tempo entre pontos adjacentes. A perda temporal e espacial causada pela generalização são calculadas separadamente e discretizadas em um intervalo de [0,1], isso possibilita dar mais peso para alguma perda em específico. Uma análise sobre a influência dos parâmetros mostra que um valor alto de k garante um maior nível de privacidade, mas em contraponto gera uma diminuição na resolução espaço-temporal. Valores baixos de t possuem um comportamento similar, pois apesar de levarem a um aumento gradual na perda de resolução ele fornece uma proteção a privacidade maior. l influência na proteção contra-ataques semânticos, valores altos também geram uma diminuição lenta na resolução espaço-temporal e um aumento no valor l irá resultar principalmente em uma garantia de diversidade de POIs. O algoritmo proposto mantém uma alta utilidade nos dados com valores de $k = 5$ e $t = 0.001$, fornecendo um nível desejável de privacidade.

Sobre a reconstrução de trajetórias tanto (NERGIZ et al., 2009) quanto (DAI et al., 2018) a utilizam para garantir a anonimidade. (NERGIZ et al., 2009) considera a anonimização de dados para dois tipos de aplicações: as sensíveis ao tempo e as sensíveis ao espaço. Ou seja: as trajetórias agrupadas precisam estar temporal e espacialmente próximas. Em cada iteração se cria um grupo vazio e se adiciona uma trajetória aleatória do grupo de trajetórias não anonimizadas. Seguindo a abordagem *multi* essa trajetória é dada como a representante do grupo e se escolhe outra trajetória próxima à trajetória representante, que são anonimizadas em uma e a resultante é a nova representante do grupo. Esse processo se repete até que o

grupo tenha k trajetórias. No fim de cada iteração um grupo é formado e removido das trajetórias não anonimizadas, e as iterações acabam quando existem menos de k trajetórias não anonimizadas. Os autores notam que a alternativa acelerada de procura de vizinhos próximos (a abordagem *fast*) da trajetória diminui a utilidade dos dados. Isso acontece pois para diminuir o número de operações realizadas não se atualiza o representante do grupo, permitindo que as $k - 1$ trajetórias mais próximas do representante sejam encontradas em uma única iteração. O processo de anonimização precisa que a correspondência ideal de pontos seja encontrada para que o custo de juntar duas trajetórias seja minimizado. O algoritmo especifica os pares de pontos entre as trajetórias e anonimiza eles entre si substituindo a região que eles representam por uma caixa delimitadora mínima que cobre a região dos dois pontos, qualquer ponto sem um par é suprimido. Ao unir mais de duas trajetórias é preciso ter um cuidado com o alinhamento delas, para lidar com isso o algoritmo identifica a trajetória que possui o menor custo em relação a todas as outras trajetórias do grupo. Em cada passo do processo se encontra a melhor correspondência entre pontos de uma trajetória não marcada do grupo com a anonimização das trajetórias marcadas. Cada marcação cria uma conexão entre os pontos. A supressão e generalização de cada ponto são aplicadas segundo a correspondência encontrada. As trajetórias são reconstruídas a partir dos grupos anonimizados, em cada caixa delimitadora mínima selecionam-se pontos atômicos para fazerem parte do trajeto. Como o artigo considera trajetórias inteiras, tanto o tamanho variável das trajetórias quanto um valor elevado de k leva a um aumento na supressão de pontos. A abordagem *multi* demonstrou uma menor distorção nos dados testados para valores altos de k comparado com a abordagem *fast*, com menos de 9% dos pontos sendo removidos.

(DAI et al., 2018) reconstrói as trajetórias após substituir seus pontos sensíveis de parada. O algoritmo considera a distribuição e a tabela semântica de atributos dos POIs, o nível de privacidade de cada usuário e a distribuição de obstáculos na região sendo anonimizada. O algoritmo possui quatro passos: nomear os atributos semânticos dos pontos de movimento e construir a árvore de taxonomia, extrair pontos de parada da trajetória de cada usuário, selecionar POI apropriados para a substituição e reconstruir a trajetória a partir da antiga trajetória alterada. Se utiliza uma tabela de atributos de POI para atribuir a cada ponto em movimento a semântica do POI mais próximo. A partir da atribuição semântica se obtém um conjunto com os atributos semânticos utilizados, cada elemento do conjunto é utilizado

para formar nós-folha da árvore, que são divididos em diferentes categorias baseado na semântica de cada nó. O valor de cada categoria é abstraído e generalizado para formar nodos superiores, esse passo se repete até que um nó raiz seja abstraído e generalizado. Após a construção da árvore se determina quais tipos de pontos em movimento precisam ser protegidos, os tipos de pontos de parada extraídos dos pontos de movimento: pontos de parada de longa duração, pontos errantes e pontos semânticos sensíveis que variam conforme as definições de privacidade do usuário. Pontos de longa duração ocorrem quando uma sequência de pontos permanece em uma certa posição acima de um limite de tempo determinado, todos os pontos em movimento dentro desse intervalo de tempo são considerados pontos de parada. Os pontos errantes são os que vagam em torno de uma área de espaço muito pequena. Um limite de distância é introduzido junto ao limite de tempo. Se considera pontos de parada pontos frequentemente acessados em um pequeno espaço acima de um determinado intervalo de tempo, a região formada por esses pontos em movimento é considerada sensível. O ponto também pode ser reconhecido como ponto de parada se o seu atributo semântico pertencer ao conjunto de atributos semânticos sensíveis definidos pelo usuário. A escolha de um POI para substituir o ponto de parada demarcado considera a velocidade do usuário utilizando as distâncias formadas pelo ponto de parada e seus pontos anteriores e posteriores para construir uma região de seleção apropriada. Também se considera uma região sobreposta entre dois pontos de parada que pode ser utilizada para inferir informações sensíveis sobre o usuário a quem essa trajetória pertence, essa sobreposição é chamada de mutação reversa da trajetória. Para evitar que essa sobreposição aconteça se utiliza de dois semi-círculos assimétricos para formar uma região apropriada de seleção (PSR), cada semi-círculo possui um raio próprio definido como metade da distância até o ponto de parada anterior ou posterior. Após a definição das regiões se procura por POIs que possuam a mesma, ou alguma similaridade, semântica que satisfaça o nível de privacidade do usuário. O conjunto de POIs candidatos mapeados é obtido, mas como os pontos de parada possuem diferentes níveis de isolamento a forma de escolher um POI varia. Se considera três categorias: pontos de parada não isolados, isolados e consideravelmente isolados. Pontos são considerados não isolados se existirem alguns POIs que pertençam a mesma categoria semântica que ele na região PSR. Pontos isolados são identificados quando na região PSR não tem nenhum outro ponto com a mesma categoria semântica que ele, nesse caso o POI escolhido possui uma categoria semântica similar. Um ponto consideravelmente isolado não possui nenhum POI em PSR

que seja apropriado para ser escolhido. Nessa situação tem se duas possibilidades: aumentar o tamanho da PSR utilizando expansão dinâmica para encontrar um candidato apropriado, ou manter esse ponto sem qualquer substituição. Para a expansão dinâmica se define um tamanho fixo para aumentar, a região irá aumentar para ambos semi-círculos. Para evitar que o aumento cause uma sobreposição de regiões com a região do ponto de parada vizinho se faz uma verificação para não escolher um POI dessa região sobreposta, essa verificação se repete até que se encontre um POI apropriado para a substituição. Na reconstrução da trajetória se substitui todos os pontos de parada sensíveis com os POIs selecionados. Os pontos de movimento no meio do caminho entre o ponto sendo substituído e o ponto de parada anterior são marcados para realizar uma alteração na rota da trajetória para que ela não fique artificial, o mesmo vale para o ponto de parada posterior ao ponto atual sendo substituído. O próximo ponto em movimento a partir dessas marcações em direção ao ponto a ser substituído é selecionado, se a condição necessária para que a alteração na trajetória aconteça a partir dele não for satisfeita se avança um ponto em direção ao ponto a ser substituído. A condição é: a diferença entre a distância desse ponto com o antigo POI com a distância desse ponto com o novo POI deve ser mínima. Após o ponto ser encontrado se gera pontos uniformes até o novo POI, e dele até o outro ponto em movimento encontrado. Esse processo de reconstrução é aplicado para cada ponto de parada até que a trajetória inteira do usuário esteja completa. Confere-se também se a trajetória reconstruída não atravessa algum obstáculo, caso isso ocorra se escolhe outro POI. Observando os resultados é possível ver que o algoritmo proposto consegue manter um nível elevado de consistência semântica entre as trajetórias originais e as reconstruídas. Os autores notam que quanto maior o número de nodos vizinhos na árvore de taxonomia menor será a consistência semântica da trajetória, a construção dessa árvore impacta na consistência semântica de todos os pontos em movimento e dos pontos de parada isolados. Um aumento no número de POIs também resulta em um aumento na consistência semântica da trajetória, já que isso leva a uma maior opção de candidatos de uma mesma categoria para cada ponto de parada. As duas abordagens sobre pontos consideravelmente isolados alcançam uma possibilidade de identificação baixa. Entre elas, a expansão dinâmica possibilita mais proteção, mas acaba tendo um maior nível de desvio da trajetória original. Os autores não testaram o algoritmo proposto em dados reais, a análise parte de testes em cima de dados sintéticos.

Apesar da grande variabilidade métodos e a empregabilidade de conceitos di-

ferentes para assegurar a privacidade das trajetórias originais os artigos no geral comparam a nova abordagem com métodos similares ao que está sendo apresentado. Este trabalho se propõe a comparar os resultados de abordagens que empregam métodos diferentes para alcançar a privacidade, analisando o tempo gasto, a qualidade dos dados e a proteção fornecida. Na tabela tabela 3 estão resumidos alguns dos métodos utilizados em cada artigo.

Artigos	Trajetórias Semânticas	Trajetórias Cruas	k-Anonimidade	l-Diversidade	t-Proximidade	Supressão	Troca de segmento	Reconstrução de Trajetórias	Regiões Generalizadas	POI
(ABUL; BONCHI; NANNI) (2008)		✓		✓		✓				
(HUO et al.) (2012)	✓			✓					✓	
(BRITO et al.) (2015)	✓		✓			✓				
(SALAS; MEGIAS; TORRA) (2018)	✓						✓			✓
(NAGHIZADE et al.) (2020)	✓						✓			✓
(GRAMAGLIA; FIORE) (2015)		✓	✓						✓	
(ZHANG et al.) (2015)	✓		✓						✓	✓
(TU et al.) (2018)	✓		✓	✓	✓				✓	✓
(NERGIZ et al.) (2009)		✓	✓			✓		✓		
(DAI et al.) (2018)	✓							✓		

Tabela 3 – Comparações entre os trabalhos relacionados.

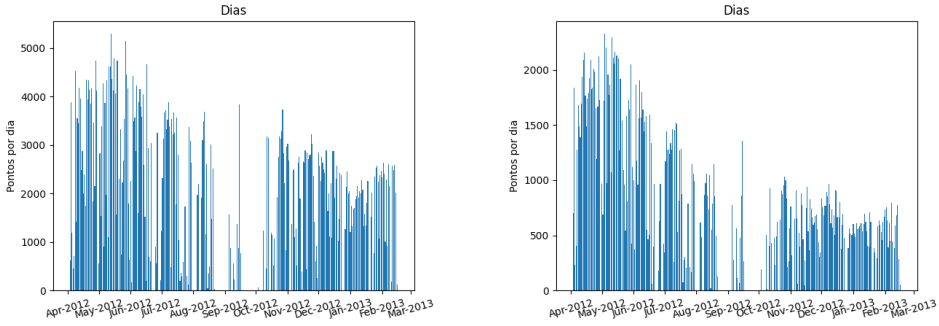
4 PROPOSTA

A proposta deste trabalho é comparar a qualidade do resultado da anonimização de trajetórias semânticas dos métodos descritos em (NAGHIZADE et al., 2020), (TU et al., 2017) e (ZHANG et al., 2015). Para os dados possuírem uma alta qualidade é necessário que eles preservem a anonimidade dos indivíduos a quem eles pertencem e também tenham uma certa utilidade, ou seja, que eles possam ser utilizados para análises estatísticas, mineração de dados, entre outros. A utilidade dos dados pode ser medida através do uso de aprendizado de máquina, avaliando a previsibilidade semântica de cada ponto de uma trajetória com o próximo ponto. Em relação à privacidade serão utilizados oito ataques apresentados em (PELLUNGRINI et al., 2017) aplicados tanto nos dados anonimizados resultantes dos três algoritmos implementados quanto nos dados originais.

4.1 CONJUNTO DE DADOS

Os conjuntos de dados utilizados no processo de anonimização são *check-ins* de Nova Iorque e de Tóquio coletados pela FourSquare (YANG et al., 2015), com os respectivos tamanhos de 227,428 e 573,703 *check-ins*. Cada registro está associado a um ID de usuário, a uma coordenada com latitude e longitude, fuso horário, um horário UTC e algumas informações semânticas sobre a região em que o registro foi coletado. No contexto deste trabalho se considera que um *check-in* é um ponto de uma trajetória.

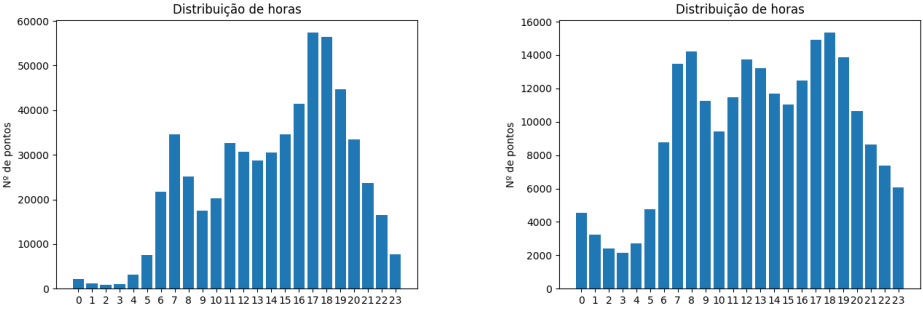
Tóquio possuí um total de 2293 usuários únicos, enquanto Nova Iorque tem 1083. A quantidade de *check-ins* por dia pode ser visto na Figura 2. As figuras apresentadas nessa seção foram geradas programaticamente na linguagem Python na produção deste trabalho a partir do conjunto de dados.



(a) Tóquio (b) Nova Iorque

Figura 2 – Quantidade de *check-ins* por dia para cada conjunto de dados.

A distribuição de horas em que os *check-ins* foram feitos pode ser observada na Figura 3.

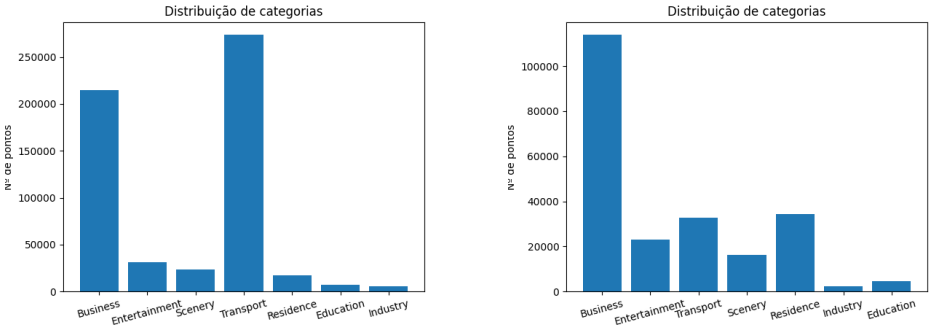


(a) Tóquio (b) Nova Iorque

Figura 3 – Distribuição de horas na qual os *check-ins* ocorreram.

Cada *check-in* de (YANG et al., 2015) possui uma informação semântica as-

sociada. No entanto, essa informação é muito específica, dificultando o processo de anonimização já que pontos com informações semânticas similares poderiam ser considerados distintos por possuírem nomes diferentes, como seria o caso de *Sushi Restaurant* e *Spanish Restaurant*. Com isso, se utilizou de base a separação feita por (TU et al., 2017) que separa as informações semânticas dos pontos em seis categorias: entretenimento, educação, cenário, negócios, indústria e residência. No caso deste trabalho se considerou uma categoria adicional: transporte. A distribuição dessas categorias pode ser observada na Figura 4.



(a) Tóquio (b) Nova Iorque

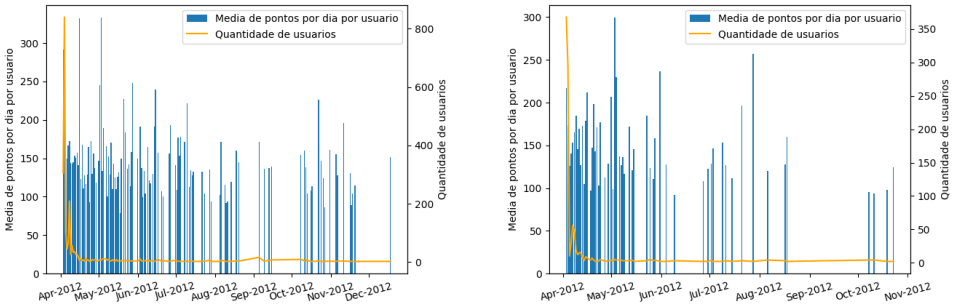
Figura 4 – Distribuição de categorias

Como o banco de dados utilizado fornece registros únicos relacionados a um identificador de usuário é necessário conectar esses registros a fim de reconstruir a trajetória correspondente. Durante a reconstrução juntam-se os pontos que possuem o mesmo identificador e dividem-se as trajetórias por dia com o intuito de manter a qualidade temporal após a anonimização. A Tabela 4 apresenta uma melhor noção das características das trajetórias totais resultantes. Acompanhando a redução no número de trajetórias conforme o aumento na restrição de mínimo de pontos é possível perceber, por exemplo, que mais da metade das trajetórias, quando divididas por dia, possuem no máximo dois pontos.

Quantidade Mínima de Pontos	Tóquio	Nova Iorque
1	200.034	94.533
2	118.303	49.825
3	73.944	27.414
4	48.928	16.205

Tabela 4 – Quantidade de trajetórias em cada banco de dados após filtrar-se as trajetórias com um determinado número mínimo de pontos

Outro ponto a se considerar sobre a qualidade dos dados utilizados é a quantidade de usuários por dia e a média de pontos das trajetórias desse dia. Como a Figura 5 demonstra, tem-se uma quantidade considerável de usuários nos dias iniciais do conjunto de dados e essa quantidade cai drasticamente com o decorrer de alguns dias. Também se observa que apesar da média de pontos por dia ser alta, não chega a uma quantidade de 50 usuários, o que pode afetar o processo de anonimização e a qualidade dos dados resultantes do algoritmo (TU et al., 2017) pois a utilização do método de k-anonimidade depende de existirem mais trajetórias para serem juntadas uma a outra.



(a) Tóquio (b) Nova Iorque

Figura 5 – Média de pontos por usuário comparado com a quantidade de usuários por dia

4.2.1 Beyond K-Anonymity: Protect Your Trajectory from Semantic Attack

O algoritmo proposto em (TU et al., 2017) procura proteger a privacidade dos usuários e manter um nível de qualidade nos dados utilizando k-anonimidade, l-diversidade e t-proximidade. A partir dos dados não tratados generalizam-se os pontos de interesse para 7 categorias: Entretenimento, Educação, Cenário, Comércio, Indústria e Residência. Essa simplificação da informação semântica presente nos POIs é importante, pois enquanto a k-anonimidade é assegurada através da junção de trajetórias, a l-diversidade e a t-proximidade são garantidas na junção de dois pontos de trajetórias diferentes, e é através da diversidade das categorias e da sua distribuição na região em que o ponto se encontra que essas garantias podem ser feitas.

Antes da execução do algoritmo têm se as seguintes premissas:

- A matriz de custo de similaridade de duas trajetórias está calculada;
- O Grafo de regiões está pronto;
- Cada ponto possui sua semântica associada com alguma categoria generalizada.

Durante a execução as duas trajetórias que possuem entre si o menor custo são escolhidas para serem juntadas. O algoritmo de junção escolhe a trajetória que possui a menor quantidade de pontos para ser juntada na outra trajetória. As regiões semânticas pelas quais as trajetórias passam e o tempo em que estavam nessas regiões é atualizado para cada ponto da nova trajetória gerada.

Esse novo ponto dessa nova trajetória é criado de forma que ele respeite os requisitos da l-diversidade e da t-proximidade, como foi mencionado anteriormente. Caso ele não respeite, procura-se no grafo pelas regiões próximas as já selecionadas e verifica-se quais delas aproximam esse ponto desse critério. Dado as características do conjunto de dados utilizado, caso o ponto sendo juntado a outro não possua nenhum vizinho a ser adicionado o critério não será respeitado com o intuito de evitar um loop infinito.

Após a trajetória resultante ser gerada, as trajetórias que a originaram são excluídas da matriz de custo e, para a nova trajetória, caso ela não respeite o critério da k-anonimidade ainda, terá seu custo calculado com todas as trajetórias restantes na matriz de custo.

4.2.2 Privacy and Context-aware Release of Trajectory Data

Em (NAGHIZADE et al., 2020) a privacidade é almejada sem que ocorra a junção de uma trajetória com outra. Para isso acontecer se considera que cada ponto que permanece um determinado δt de tempo em uma distância δd é um ponto de parada. Cada ponto de parada possui um indicador do nível de privacidade do local, que permite que caso uma trajetória tenha um ponto de parada com um rigor de privacidade alto ele possa ser alterado para outro ponto de parada realístico que possua um rigor de privacidade menor.

Antes da execução do algoritmo têm se as seguintes premissas:

- O indicador do nível de privacidade de cada trajetória para cada POI já está configurado.

- Todos os POIs de todas as trajetórias já foram identificados.

Para o algoritmo conseguir imitar pontos de parada mais realísticos nessa troca ele considera as outras categorias identificadas em outros pontos da trajetória e procura alterar esse ponto de parada expositivo para um que seja diferente das categorias já presentes na trajetória e que não seja tão sensitivo.

O ponto de interesse a ser alterado pode ser um presente na própria trajetória, caso contrário será necessário que o algoritmo procure algum POI próximo do que está para ser substituído. A busca utiliza as propriedades de uma elipse para diminuir o tamanho de busca. O primeiro caso é considerado troca de parada, mantendo assim o caminho original da trajetória. O segundo se refere como deslocamento de parada e pode causar alterações tanto temporais quanto espaciais.

O artigo apresenta ainda outra abordagem, mas apenas esta que foi descrita ("Troca Flip-Flop") será considerada neste trabalho. Por uma questão de simplicidade se considerou as mesmas categorias semânticas de (TU et al., 2017) para os

POIs, isso foi necessário para associar o nível de privacidade de um usuário durante a substituição de um POI.

4.2.3 An Efficient Method on Trajectory Privacy Preservation

A abordagem apresentada em (ZHANG et al., 2015) utiliza a k-anonimidade se baseando nos POIs. De forma inicial se extraem os POI existentes nas trajetórias, criam-se as ROIs a partir de *clusters* de POIs e as trajetórias são particionadas com base nas ROIs. Por fim, as trajetórias são anonimizadas em cada conjunto de k-anonimização separado e então publicadas.

Os pontos de interesse possuem uma aproximação diferente de (TU et al., 2017), nesse caso pontos são considerados POI se: são pontos de parada, pontos de virada, pontos de início ou fim da trajetória. Pontos de virada ocorrem quando o ângulo entre dois segmentos adjacentes passa de um certo limiar definido previamente.

POIs que estão em um tempo-espaço próximo são colocados em um mesmo *cluster*. Para cada ponto de um *cluster* confere-se se o número de vizinhos é menor que um limiar de densidade, e caso positivo esse ponto é marcado como um candidato de ruído preliminar, caso contrário o ponto é considerado um ponto central e ele precisa construir um novo *cluster*.

Depois de todos os pontos no conjunto serem visitados, aqueles marcados como candidatos de ruído são processados. Verifica-se se eles possuem algum vizinho e se possuírem são adicionados no *cluster* mais próximo, caso contrário serão removidos. Esses clusters são as ROIs. Para alcançar a k-anonimidade é necessário que cada grupo de trajetórias tenha entre k e $2k - 1$ trajetórias. Trajetórias que ao final não foram adicionadas em nenhum grupo são adicionadas em qualquer grupo que tenha pelo menos uma trajetória com uma similaridade maior que um valor arbitrário α. Grupos que possuírem menos que k trajetórias são removidos.

4.3 RISCO DE EXPOSIÇÃO

O Risco de exposição de um indivíduo é a sua probabilidade de ser reidentificado em um conjunto de dados móveis. A aplicação de ataques de reidentificação antes da publicação auxilia a averiguar a melhor troca entre a qualidade

dos dados e a privacidade dos usuários a quem esses dados pertencem.

Ataques de re-identificação partem do pressuposto de que o atacante possui um conhecimento prévio acerca de um indivíduo. Os elementos associados ao conhecimento prévio de ataques de re-identificação aplicados a conjunto de dados móveis estão associados a um local e um horário.

Definição 1 (**Configuração de Conhecimento Prévio**). *Dado um conhecimento prévio B, denota-se $B_k \in B = \{B_1, B_2, ..., B_n\}$ uma configuração de conhecimento prévio específica, onde k representa o número de elementos de B conhecidos pelo atacante. Um elemento $b \in B_k$ é definido como uma instância de configuração de conhecimento prévio.*

Na definição 1, retirada de (PELLUNGRINI et al., 2017), k indica quantos elementos fazem parte do conhecimento prévio do atacante. O conjunto B representa todas as possíveis configurações de conhecimento prévio que um atacante pode possuir. Os ataques baseados nessa definição alteram apenas os elementos sendo considerados como conhecimento prévio, podendo ser utilizado apenas a informação de lugar, mas também conseguindo associar um horário em que o indivíduo estava nesse lugar.

Definição 2 (**Probabilidade de re-identificação**). *Dado um ataque, a função de $matching(D, b)$ indica se um registro $d \in D$ corresponde a uma configuração de instância de conhecimento prévio $b \in B_k$, e uma função $M(D, b) = \{d \in D | matching(d, b) = True\}$, é definido a probabilidade de re-identificação de um indivíduo u no conjunto de dados D como:*

$$PR_D(d = u|b) = \frac{1}{|M(D, b)|,} \tag{4.1}$$

isto é, a probabilidade de associar um registro $d \in D$ a um indivíduo u, dada uma instância $b \in B_k$.

Dadas as eq. (4.1) e definição 3, retiradas de (PELLUNGRINI et al., 2017), tem-se que o risco de re-identificação de um usuário é a mais alta probabilidade de re-identificação das suas instâncias encontrada considerando todas as ocorrências no conjunto de dados.

Definição 3 (**Risco de re-identificação ou risco de privacidade**). *O risco de re-identificação (ou risco de privacidade) de um indivíduo u dado uma configuração de conhecimento prévio* B_k *é a sua probabilidade máxima de re-identificação.* $Risk(u, D) = maxPR_D(d = u|b)forb \in B_k$. *E o risco de re-identificação possui um limite inferior* $\frac{|D_u|}{|D|}$ *(uma escolha aleatória em D), e* $Risk(u, D) = 0if u \notin D$.

4.4 APRENDIZADO DE MÁQUINA

Serão utilizados cinco algoritmos de aprendizado de máquina supervisionado: árvores de decisão, redes neurais artificiais, máquinas de vetores de suporte, *k-nearest neighbours* e *naive bayes*. Os algoritmos serão executados separadamente tanto para os dados anonimizados com cada método implementado quanto para os não tratados.

O objetivo é verificar o quão previsível a próxima categoria semântica de um ponto é em relação ao próximo para avaliar a anonimização na parte semântica, fazendo uma análise em relação à privacidade dos usuários e a utilidade dos dados. Esses cinco algoritmos foram escolhidos de forma arbitrária para ter diferentes combinações de técnicas e de características.

5 IMPLEMENTAÇÃO DOS ALGORITMOS

5.1 PRÉ-PROCESSAMENTO

Antes do processo de anonimização dos dados há um pré-processamento para separar as trajetórias. Cada algoritmo possui especificações particulares de entrada, mas de forma geral todos os dados foram padronizados para o formato *Trajectory*, que por sua vez é uma lista de *Point*. Ambos formatos podem ser observados na Listagem 1. Cada *check-in* possui as informações apresentadas na Tabela 5 e ao final desta primeira etapa de pré-processamento os dados estarão organizados conforme a estrutura definida na listagem 1.

Campo	**Exemplo**	**Descrição**
userId	123	Identificador do usuário
venueId	49bbd6c0f964a520f4531fe3	Identificador do local
venueCategoryId	4bf58dd8d48988d127951735	Identificador de categoria do local
venueCategory	Arts & Crafts Store	Nome da categoria
latitude	40.71981038	-
longitude	-74.00258103	-
timezoneOffset	-240	Fuso horário
utcTimestamp	Tue Apr 14 09:15:30 +0000 2011	Marca temporal

Tabela 5 – Campos do conjunto de dados.

```
@dataclass
class Trajectory:
    points: list[Point]

@dataclass
class Point:
    name: str
    uid: str
    venue_id: set[str]
    category: str
    lat: float
    lon: float
    timestamp: datetime
    duration: timedelta
```

Listing 1 – Estruturas das classes Trajectory e Point.

Em seguida as trajetórias são divididas por dia e nesse momento é possível determinar um número mínimo de pontos por trajetória. Após isso, o algoritmo de (TU et al., 2017) tem a duração de cada ponto contabilizada. Isso ocorre comparando um ponto de uma trajetória de um dia com o próximo ponto da mesma, caso os dois pontos possuam o mesmo *venueId* se considera que o indivíduo permaneceu do tempo de início do primeiro ponto até o tempo de início desse segundo ponto.

Tanto (TU et al., 2017) quanto (NAGHIZADE et al., 2020) passam por uma generalização da informação semântica, alterando as estruturas de *Trajectory* e *Point* para *SemanticTrajectory* e *SemanticPoint* respectivamente. As estruturas podem ser observadas na listagem 2.

```python
@dataclass
class SemanticPoint:
    uid: str
    category: PoiCategory
    latitude: float
    longitude: float
    timestamp: datetime
    duration: timedelta
    type: str = ""

class SemanticTrajectory:
    points: list[SemanticPoint]
    n: int = 1
```

Listing 2 – Estruturas das classes *SemanticTrajectory* e *SemanticPoint*.

A generalização semântica dos dados pode ser vista na Tabela 6, essa generalização foi baseada na utilizada por (TU et al., 2017) com o incremento da categoria *Education*. Qualquer valor de *venueCategory* que esteja presente na coluna Termos é traduzido para o seu respectivo valor na coluna Categoria. Apesar do algoritmo (ZHANG et al., 2015) não utilizar a informação semântica proveniente dos dados a generalização também é feita para esse método no final de sua anonimização para que o resultado possa ser avaliado com aprendizado de máquina.

Categoria	Termos
Business	Shop, Store, Restaurant, Bakery, Wash, Embassy, Ramen, Diner, Salon, Place, Steakhouse, Market, Joint, store, Food, Service, Bar, Café, Cafe, Office, Bank, Mall, Newsstand, Fair, Tea, City, Gastropub, Studio, Bodega, Rental, Dealership, Photography Lab, Medical Center, Tattoo.
Industry	Factory, Military, Distillery, Government, Harbor, Facility, Winery, Brewery.
Residence	Residential Building, Building, Shelter, Neighborhood, Home, Hotel, Housing, House.
Scenery	Scenery, Scenic, Park, Outdoors, Garden, Museum, Castle, River, Cemetery, Temple, Synagogue, Church, Shrine, Historic, Mosque, Planetarium, Spot, Rest Area, Plaza, Spiritual, Campground.
Education	School, College, Student, University.
Entertainment	Music, Movie, Playground, Arcade, Art, Entertainment, Gym, Nightlife, Spa, Pool, Library, Aquarium, Beach, Zoo, Bowling, Theater, Athletic, Casino, Comedy, Stadium, Concert, Convention, Ski, Racetrack.
Transport	Train, Bike, Airport, Ferry, Station, Road, Moving, Transport, Subway, Bridge, Travel, Taxi, Light Rail.

Tabela 6 – Categoria e respectivos termos.

Há, para o algoritmo (ZHANG et al., 2015), uma etapa anterior a todas essas que é o processo de filtragem dos dados. Neste processo se remove, para cada indivíduo, pontos considerados como ruído ou com valores discrepantes. Também se configura uma velocidade máxima aceitável para os pontos da trajetória, ou seja, pontos que possuem uma grande distância em relação ao ponto anterior em um período muito curto de tempo são desconsiderados.

5.2 BEYOND K-ANONYMITY: PROTECT YOUR TRAJECTORY FROM SEMANTIC ATTACK

A implementação do método proposto por (TU et al., 2017) possui três etapas principais: a construção do grafo de regiões, a construção da matriz de custo e por fim a anonimização do conjunto de dados.

5.2.1 Construção do Grafo

Após o pré-processamento dos pontos do conjunto de dados é a construção do grafo de regiões que o algoritmo usa durante a anonimização. O grafo foi organizado de forma que seus vértices sejam um conjunto de *Regions*. Também foi adicionado um dicionário para armazenar a distribuição das categorias de todo o conjunto de dados, esse valor é calculado após todas as trajetórias e seus pontos serem processados. O *default_factory* foi utilizado em alguns atributos para inicializá-los com suas respectivas estruturas de dados vazias. A organização do código pode ser observada em Listagem 3.

```python
@dataclass
class Graph:
    vertices: set[Region] = field(default_factory=set)
    poi_distribution: dict[PoiCategory, float] =
    ↪  field(default_factory=list)

@dataclass
class Region:
    id: int
    center_point: tuple[float, float]
    area: int
    categories: dict[PoiCategory, int] = field(default_factory=dict)
    neighbours_id: set[int] = field(default_factory=set)
```

Listing 3 – Estruturas das classes Graph e Region.

De início se considera que cada ponto é uma *Region*. Ao final dessa etapa se chama uma função para unir regiões que estejam próximas o suficiente, toma-se uma região como referência e confere-se todas as outras regiões para verificar se elas estão dentro de sua área. Para agilizar o processo todas as regiões encontradas são juntadas de uma vez só. O *center_point* resultante é a média de todos os *center_points* das regiões consideradas e a quantidade de categorias nessa região é atualizada. Durante esse processo também se atualiza as regiões vizinhas de cada região.

Junto da construção do grafo altera-se a estrutura utilizada para representar uma trajetória e um ponto como pode ser observado em Listagem 4. Para o ponto se mantém apenas informações essenciais: a marca temporal, o tempo de duração do ponto e o id da região em que ele estava. Como o algoritmo trabalha com a junção de trajetórias, se mantém uma referência de quais foram os *ids* de usuários e quantos já foram juntados. Cada trajetória possui um id único como referencial para ser encontrada na matriz de custo.

```python
@dataclass
class TuPoint:
    utc_timestamp: datetime
    duration: timedelta
    region_id: list[int]


@dataclass
class TuTrajectory:
    uid: list[str]
    id: int
    points: list[TuPoint]
    n: int = 1
```

Listing 4 – Estruturas das classes TuTrajectory e TuPoint.

Após o processo de junção de regiões no grafo é necessário atualizar o *id* dos vizinhos de cada região e também o *id* da região de cada ponto.

5.2.2 Construção da matriz de custo

Para a construção da matriz de custo cada trajetória será comparada com outra existente no conjunto de dados. Duas trajetórias que possuírem mais uma diferença temporal superior a θ_T^m uma com a outra não terão seu custo calculado e portanto não possuem possibilidade de junção. O custo é calculado ponto a ponto, tendo metade de seu valor atribuído a perda temporal de junção e a outra metade a perda espacial.

A fórmula para o cálculo da perda espacial e temporal segue o apresentado em (TU et al., 2017). Seguindo as eqs. (5.1) e (5.2), tem-se que d é a duração, sendo d_c a duração resultante da junção dos pontos e d_a e d_b as suas durações atuais. n_i e n_j indicam quantas vezes a trajetória a que esse ponto pertence já foi juntada a outras trajetórias, sendo i associado ao d_a e j ao d_b. Por fim θ_T^m é um valor definido como o limite de diferença de tempo máximo aceitável.

$$\theta_T^*(t_a, d_a, t_b, d_b) = \frac{(d_c - d_a)n_i + (d_c - d_b) * n_j}{n_i + n_j} \tag{5.1}$$

$$\theta_T(t_a, d_a, t_b, d_b) = min\{\frac{\theta_T^*(t_a, d_a, t_b, d_b)}{\theta_T^m}, 1\} \tag{5.2}$$

Para a perda espacial, nas eqs. (5.3) e (5.4), S_l representa a área espacial da localização l. De forma similar, S_{l_c} é referente a soma das áreas dos dois pontos e S_{l_a} e S_{l_b} a área total de cada ponto. O cálculo da área é a soma de todas as regiões que o ponto está associado. θ_L^m representa o limite máximo de área aceitável.

$$\theta_L^*(l_a, l_b) = \frac{(S_{l_c} - S_{l_a})n_i + (S_{l_c} - S_{l_b})n_j}{n_i + n_j} \tag{5.3}$$

$$\theta_L(l_a, l_b) = min\{\frac{\theta_L^*(l_a, l_b)}{\theta_L^m}, 1)\} \tag{5.4}$$

Com isso, tem-se na eq. (5.5) a fórmula para calcular o custo de junção de dois pontos. Sendo ω_T e ω_L os fatores de normalização e p_a e p_b os pontos das duas trajetórias sendo analisadas.

$$\theta(p_a, p_b) = \omega_T\theta_T(t_a, d_a, t_b, d_b) + \omega_L\theta_L(l_a, l_b) \tag{5.5}$$

5.2.3 Anonimização

Durante o processo de anonimização o algoritmo fica juntando trajetórias enquanto tiver mais de duas trajetórias não anonimizadas ou enquanto o tamanho da matriz de custo for superior a dois. Essa segunda restrição foi adicionada pois nem todas as trajetórias tem seu custo calculado uma a outra, isso acontece quando duas trajetórias possuem uma diferença temporal superior a θ_T^m.

As trajetórias com o menor custo encontradas na matriz de custo são escolhidas para serem juntadas. No processo de junção se identifica a trajetória com a maior quantidade de pontos e a com a menor quantidade. Para cada ponto da maior trajetória se encontra qual o ponto da menor com o custo de junção mais baixo, os pontos da menor que forem juntados com outros são identificados em uma lista. Ao

final deste processo, todos os pontos não juntados da menor trajetória são juntados com o ponto modificado de menor custo da mesma.

Na função de junção de dois pontos se calcula a nova marca de tempo e duração do novo ponto. Também se confere se as regiões dos pontos são a mesma ou se são vizinhas, caso contrário se seleciona, a partir do algoritmo de *Dijkstra* e com o auxilio do grafo construído, as regiões intermediárias entre esses dois pontos. Em seguida se verifica a diversidade dos pontos: para cada categoria presente nas regiões que conectam ambos os pontos selecionados se considera 1, a diversidade é a soma de todas as categorias. Para a proximidade, o cálculo é feito seguindo a eq. (5.6) onde X_u é a distribuição de uma categoria nas regiões dos pontos selecionados e Y_u a distribuição de uma categoria em todo o grafo.

$$\delta_t^r = \sum_{u=1}^{7} X_u log \frac{X_u}{Y_u} \tag{5.6}$$

Se a diversidade não possuir um valor maior que o δl e a proximidade não possuir um valor menor que δt o algoritmo irá procurar vizinhos das regiões sendo consideradas e irá adicionar a região vizinha que mais agrega ao limiar em questão. Caso não existam mais vizinhos para serem adicionados o código aceita o estado atual de diversidade e proximidade, essa adição é feita com o intuito de evitar um *loop* infinito.

Após a junção das trajetórias ser finalizada, as trajetórias que a originaram são excluídas da matriz de custo e do conjunto de trajetórias. Caso a trajetória resultante não respeite o critério de k-anonimidade ela terá seu custo calculado para todas as outras trajetórias restantes e será adicionada ao conjunto de trajetórias não anonimizadas. Se a trajetória alcançar o critério de anonimidade ela é adicionada no conjunto de trajetórias anonimizadas.

5.3 PRIVACY AND CONTEXT-AWARE RELEASE OF TRAJECTORY DATA

Para executar a implementação do método de (NAGHIZADE et al., 2020) há duas etapas anteriores necessárias: a de geração de configurações de usuário e a identificação dos pontos de parada das trajetórias.

5.3.1 Geração de configurações de usuário

Em (NAGHIZADE et al., 2020) o usuário é responsável por classificar os POI. Para conseguir reproduzir este cenário adicionou-se a cada trajetória pré-processada um dicionário que relaciona a categoria de um POI a um valor aleatório, indo de 0.1 a 1.0, definido em sua geração.

5.3.2 Identificação de pontos de parada

Nesta etapa se itera por cada trajetória para identificar os pontos de parada. Se considera que a trajetória é segmentada em pontos de parada e pontos de movimento. Pontos de parada são identificados a partir de um conjunto de pontos que possua uma distância de um para outro menor que δd e que possua uma duração temporal maior que δt.

A partir dessa iteração se altera a estrutura de representação da trajetória para *Segmented*, e os pontos passam a ser estruturas denominadas *Stop* e *Move*. Pontos pertencentes a estrutura *Move* possuem as coordenadas, um horário de início e um horário de fim. Pontos de *Stop* também possuem esses dados mas tem a mais a semântica de onde o individuo parou e quão sensitivo esse ponto é de acordo com as configurações da trajetória que o originou. As estruturas podem ser observadas na listagem 5.

```
class Stop:
    locations: list[tuple[float, float]]
    start: datetime
    end: datetime
    semantic: PoiCategory
    sensitivity: float = None


@dataclass
class Move:
    locations: list[tuple[float, float]]
    start: datetime
    end: datetime


class Segmented:
    uid: str
    points: list[Stop | Move]
    privacy_settings: dict[PoiCategory, float] =
    ↪  field(default_factory=list)
    length: float = 0
```

Listing 5 – Estruturas das classes *Segmented*, *Move* e *Stop*.

A sensibilidade de um ponto é processada ao fim da identificação de pontos de parada. Isso ocorre pois é necessário ter a duração total da trajetória para conseguir ter o valor de sensibilidade do ponto. O cálculo segue a eq. (5.7), sendo r_p o valor atribuído a sensibilidade do ponto àquela categoria nas configurações iniciais, d_t a duração do ponto de parada e d_s a duração total da trajetória.

$$r_s = r_p^{\frac{d_t - d_s}{d_t}} \tag{5.7}$$

Após essa alteração de estrutura se cria uma lista com todos os POIs, selecionando todos os pontos das trajetórias que pertençam ao tipo *Stop*.

5.3.3 Anonimização

O processo de anonimização ocorre de forma individual para cada trajetória. De início, procura-se identificar uma sub-trajetória sensível percorrendo os segmentos da trajetória e verificando se a sensibilidade dos pontos do tipo *Stop* satisfazem as configurações de privacidade da trajetória em questão.

Caso essa condição não seja satisfeita, esse ponto é sinalizado como o início de uma sub-trajetória sensível. O próximo *Stop* sensível encontrado irá sinalizar o fim da sub-trajetória encontrada e o início de outra, esse ponto é considerado como parte da sub-trajetória. Os pontos de movimento *Move* só são considerados como parte da sub-trajetória após a identificação de um primeiro ponto *Stop*. Com isso, se considera que as sub-trajetórias sensíveis encontradas sempre terão como ponto inicial um ponto do tipo *Stop*.

Com as sub-trajetórias sensíveis identificadas, a próxima etapa é procurar por POIs menos sensíveis para substituí-las. Cada sub-trajetória tem o objetivo de substituir o primeiro ponto *Stop* por outro menos sensível.

A partir da trajetória ocorre uma busca em si para encontrar um POI com uma sensibilidade menor que a sensibilidade atual do ponto *Stop* a ser alterado. Os POIs encontrados nessa busca são armazenados em uma lista e o menos sensível é escolhido para ser o substituto, com o POI encontrado a função *stop_ replacement* é chamada. Caso nenhum seja encontrado se chama a função *stop_ displacement* que é responsável por procurar um POI substituto em todos os POIs identificados a partir das trajetórias.

Na substituição de dois POIs é necessário trocar o tempo de permanência de um POI com o de outro, impactando na duração do primeiro e do segundo POI e também na marca temporal de início do segundo. A função *stop_ replacement* é responsável por executar essa consideração temporal, recebendo como parâmetros a sub-trajetória e qual POI dessa sub-trajetória será alterado. Importante notar que essa alteração impacta no horário inicial e final dos pontos que se encontram entre o POI escolhido e o POI a ser substituído.

No segundo caso, no *stop_ displacement*, se constrói uma elipse tendo como foco o primeiro e o último ponto da sub-trajetória sendo alterada. A partir dos limites da elipse ocorre uma busca que procura todos os POI que estejam dentro de sua região. Se nenhum POI for encontrado então a área da elipse é aumentada, o

processo se repete até um POI ser encontrado ou enquanto o eixo maior for menor que metade do tamanho da trajetória total, sendo o tamanho da trajetória a sua distância total percorrida.

A partir dos POI encontrados se escolhe a partir das configurações do usuário o POI com a menor sensibilidade. Como o POI escolhido nessa situação não faz parte do trajeto original da sub-trajetória é necessário formar um novo trajeto. Para isso se utiliza da API *Open Source Routing Machine* (OSRM), se faz duas requisições: uma do início da sub-trajetória até o novo POI, e outra desse POI até o final da sub-trajetória. A resposta da API é uma lista com várias coordenadas geográficas, esse resultado é considerado como um segmento de *Move*. Esse novo trajeto é passado para a função *stop_ replacement* que irá fazer a substituição de POI.

Trajetórias que não possuem sub-trajetórias sensíveis identificadas não passam pelo processo de anonimização.

5.4 AN EFFICIENT METHOD ON TRAJECTORY PRIVACY PRESERVATION

O método descrito em (ZHANG et al., 2015) consiste de três etapas: a extração de POIs, a criação de *clusters* e a troca de POIs sensíveis entre trajetórias de um mesmo *cluster*.

5.4.1 Extração de POI

POIs consistem em pontos de parada, pontos de virada e pontos no início e ao fim da trajetória. Com essas novas informações a serem consideradas se utiliza uma estrutura PoI, apresentada na listagem 6 para armazenar as informações dos POI encontrados. *id*, *loc* e *t* são respectivamente o identificador único do POI, a sua localização em coordenadas geográficas e a sua marca temporal. *semantic* é a categoria semântica associada a esse POI e *neighbours* é o conjunto de *id*s de outros POI próximos a si. O atributo *semantic* não é utilizado durante o método, mas a sua informação é necessária para a avaliação realizada com aprendizado de máquina realizada posteriormente.

```python
@dataclass
class PoI:
    id: int
    loc: tuple[float, float]
    t: datetime
    semantic: PoiCategory
    neighbours: set[int] = field(default_factory=set)
```

Listing 6 – Estrutura de um PoI

A identificação dos POIs é feita trajetória a trajetória. Como os pontos de início e fim são adicionados diretamente resta procurar por pontos de parada e pontos de virada. Após indicar qual é o primeiro ponto sendo analisado a primeira coisa a se fazer é encontrar um ponto candidato na trajetória, para isso basta que o ponto sendo considerado possua uma distância espacial maior que o limiar *min_ dist*. Caso nenhum ponto seja encontrado se considera que a trajetória atual não possui nenhum POI.

Caso contrário, verifica-se a diferença temporal entre o ponto analisado e o ponto candidato, se a diferença temporal for maior que *min_ stay_ time* o ponto analisado é considerado um POI. Se o ponto candidato não for o ponto em sequência a partir do ponto analisado as coordenadas geográficas do POI serão a média das coordenadas do ponto analisado e do candidato.

Se a diferença temporal entre os dois pontos não alcançar o limiar será calculado o ângulo entre o ponto analisado, o candidato e o próximo ponto a partir do candidato. Caso esse ângulo seja maior que *min_ angle* então o ponto analisado é considerado um POI. No final da iteração o ponto candidato passa a ser o ponto analisado e a busca por um novo ponto candidato acontece novamente, esse processo ocorre até que se chegue ao final da trajetória.

Durante esse processo ocorre uma modificação da estrutura utilizada para as trajetórias, deixando de ser uma *Trajectory* e passando a ser uma *ZhangTrajectory*. Essa modificação ocorre para que os POIs, e posteriormente os ROIs também, na qual a trajetória faz parte sejam facilmente identificados. A estrutura pode ser vista na listagem 7.

```python
@dataclass
class ZhangTrajectory:
    uid: str
    points: list[Point | PoI]
    pois: set[int]
    rois: set[int] = field(default_factory=set)
```

Listing 7 – Estrutura de *ZhangTrajectory*.

5.4.2 Criação de ROIs

Antes de começar a agrupar POIs em ROIs se calcula a distância de um POI a outro para todos os POIs identificados, se um POI estiver a uma distância espacial menor ou igual a *spatial_ radius* e a uma distância temporal menor ou igual a *temporal_ radius* os POIs são considerados vizinhos.

De início, cria-se uma lista vazia para armazenar os POIs já visitados, toda vez que um POI for visitado o seu identificador é adicionado a essa lista. Uma ROI é criada ao redor de um ponto caso ele possua uma quantidade de vizinhos maior que *density_ threshold*, se um POI não atingir esse limiar ele é considerado um candidato a ruído. Após todos os POI serem processados, POIs com ao menos um vizinho são adicionados ao ROI mais próxima de si.

A estrutura de uma ROI utilizada é composta por um identificador único, o identificador do POI inicial da região e os identificadores dos POIs que fazem parte dessa ROI.

A criação de uma ROI acontece a partir de um POI, dado esse POI central se considera os vizinhos desse POI analisando somente os POIs que não foram visitados ainda. Se o ponto vizinho sendo analisado tem uma quantidade de vizinhos maior que *density_ threshold* então se adiciona eles na lista de vizinhos a serem considerados.

Após esse momento, se o ponto sendo analisado não está em uma ROI ele é adicionado a ROI sendo que está sendo criada. Caso nenhum vizinho novo seja encontrado se finaliza a criação da ROI.

5.4.3 Criação de grupos

Com as ROIs definidas e também pontos considerados como ruídos descartados falta atualizar as trajetórias com essas informações. Para cada POI de cada trajetória é verificado em qual ROI esse POI pertence. Se nenhum ROI for encontrado então esse POI é excluído da trajetória.

Para cada trajetória se calcula a similaridade dela com as outras, trajetórias que possuírem similaridade igual a 1 são agrupadas junto. Grupos somente são considerados se possuírem mais de 1 trajetória em si. O cálculo da similaridade entre duas trajetórias pode ser observado na eq. (5.8) sendo R_1 e R_2 as ROIs de cada trajetória.

$$Sim(tr_1, tr_2) = \frac{|R_1 \cap R_2|}{|R_1 \cup R_2|} \tag{5.8}$$

Para calcular a similaridade entre uma trajetória e um grupo de trajetórias foi feito um somatório como pode ser observado na eq. (5.9).

$$Sim(group, tr_1) = \sum_{i=0}^{|group|} Sim(tr_1, tr_i) \tag{5.9}$$

Trajetórias que não forem agrupadas tem sua similaridade calculada a grupos existentes, se a similaridade a esse grupo for maior que um valor α ela é adicionada a esse grupo. Após isso verifica-se o tamanho dos grupos formados, grupos que possuírem menos de k trajetórias são desconsiderados. Por fim, cada POI de cada trajetória é trocado por outro POI da mesma ROI que ele pertence.

6 EXPERIMENTOS

Neste capítulo estão descritas as configurações utilizadas na coleta dos resultados. Os experimentos foram realizados em um computador com AMD Ryzen 5 5600X 6-Core e 16GB RAM DDR4 2400MHz.

6.1 ALGORITMOS DE ANONIMIZAÇÃO

O tamanho do conjunto de dados utilizado para cada algoritmo não foi o mesmo devido ao tempo disponível e a demora no processo de anonimização. Importante notar que apesar da quantidade de dados processada no algoritmo (TU et al., 2017) se anonimizou no período de 1 semana 754 trajetórias para o conjunto de dados de Nova Iorque e 46 para o de Tóquio.

Algoritmo	Tóquio	Nova Iorque
(NAGHIZADE et al., 2020)	573704	227429
(ZHANG et al., 2015)	71712	56856
(TU et al., 2017)	286852	113715

Tabela 7 – Quantidade de pontos utilizado para cada método a depender do conjunto de dados.

Os valores utilizados para os parâmetros dos algoritmos (ZHANG et al., 2015), (NAGHIZADE et al., 2020) e (TU et al., 2017) estão expostos respectivamente nas tabelas 8 a 10.

Para (NAGHIZADE et al., 2020) apenas se considerou trajetórias que possuíssem um mínimo de 2 pontos por dia, em (TU et al., 2017) o mínimo foi 3. Em relação a (ZHANG et al., 2015) não foi exigido um mínimo de pontos, pois se realiza uma filtragem de velocidade de trajetória no início de seu pré-processamento.

Parâmetros	Valor	Unidade
min_angle	30	Graus
min_dist	200	Metros
min_stay_time	20	Minutos
spatial_radius	200	Metros
temporal_radius	20	Minutos
alpha	0	-
k	2	-

Tabela 8 – Valores utilizados para os parâmetros no método proposto em (ZHANG et al., 2015)

Parâmetros	Valor	Unidade
dist_threshold	200	Metros
temporal_threshold	20	Minutos

Tabela 9 – Valores utilizados para os parâmetros no método proposto em (NAGHIZADE et al., 2020)

Parâmetros	Valor	Unidade
k	2	-
l	4	-
t	0.01	-

Tabela 10 – Valores utilizados para os parâmetros no método proposto em (TU et al., 2017)

Em (TU et al., 2017) também se considerou 150 como o tamanho inicial de uma região.

6.2 ATAQUES

Para executar os ataques foi necessário processar os dados anonimizados. Cada trajetória construída foi separada em uma lista de pontos com as informações de identificação do usuário inicial, latitude, longitude e marca temporal.

Espera-se que os dados estejam no formato *TrajDataFrame*, para isso é necessário passar a lista de pontos indicando em qual coluna se encontra as informações separadas na etapa anterior. De exemplo, um ponto de uma trajetória anonimizada que juntou os identificadores 1, 2, 3 e 7, possui como coordenada central de sua região os valores 10 de latitude e 20 de longitude, e possui uma marca temporal de 10 horas da manhã do dia 12 de maio de 2012 seria padronizado como demonstra a Tabela 11. Dada a criação dessa lista, basta indicar para o construtor do *TrajDataFrame* que as colunas 0, 1, 2 e 3 são respectivamente *user_ id*, latitude, longitude e *datetime*.

Identificador	Latitude	Longitude	Marca Temporal
1	10	20	Sat May 12 10:00:00 +0000 2012
2	10	20	Sat May 12 10:00:00 +0000 2012
3	10	20	Sat May 12 10:00:00 +0000 2012
7	10	20	Sat May 12 10:00:00 +0000 2012

Tabela 11 – Exemplo de padronização de pontos anonimizados para o formato esperado de *TrajDataFrame*.

Para os dados obtidos a partir do método de (TU et al., 2017), as coordenadas de latitude e longitude de um ponto são dadas pela média de todas as regiões na qual esse ponto é considerado fazer parte. O identificador inicial de cada usuário é mantido atrelado a trajetória e a marca temporal é um atributo que pode ser acessado.

Para (NAGHIZADE et al., 2020) ocorre uma construção temporal para os pontos já que foram considerados segmentos de movimento e de parada. O tempo de duração de cada segmento é calculado e esse valor é dividido pela quantidade de coordenadas presentes. Para cada coordenada se soma esse tempo de duração com o tempo inicial do segmento, obtendo assim a marca temporal desse ponto e

atualizando o valor de referência de início. Os valores referentes ao identificador de usuário inicial e a marca temporal são atributos da trajetória.

No caso do método de (ZHANG et al., 2015), todos as informações necessárias estão contidas tanto nos pontos quanto nos POI presentes nas trajetórias. Basta organizar as informações no formato exigido pelo *TrajDataFrame*.

A análise de risco realizada separadamente por cada ataque considera um conhecimento prévio de tamanho 2. Esse tamanho especifica quantos lugares o atacante conhece sobre o indivíduo sendo analisado. A partir da padronização dos dados se chama a rotina de avaliação de risco disponibilizada pela biblioteca *scikit-mobility* (PAPPALARDO et al., 2022).

6.3 APRENDIZADO DE MÁQUINA

Para avaliar a previsibilidade semântica da categoria de um ponto foi realizado um pré-processamento. Os dados anonimizados de cada algoritmo foram divididos em um conjunto para o treinamento do classificador e outro para o teste, a divisão foi respectivamente 25% e 75%.

Após essa divisão foi construído duas listas relacionadas, sendo a primeira lista com a categoria atual e outra com a próxima categoria. Importante constar que as categorias utilizadas nesse aspecto foram a versão generalizada.

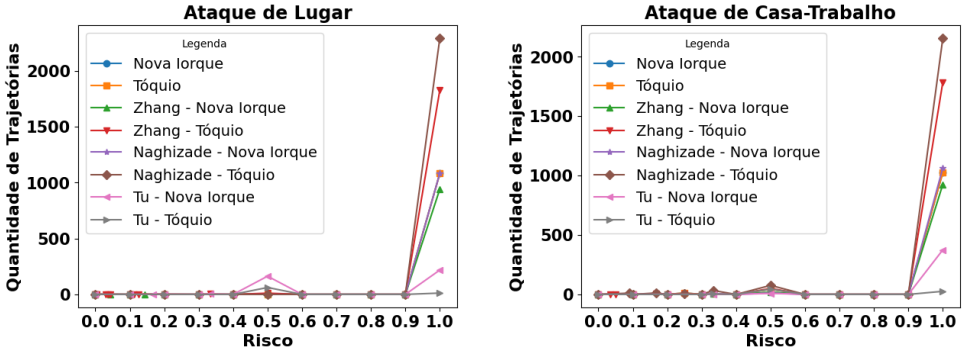
Os algoritmos de aprendizado de máquina utilizados foram: árvores de decisão, redes neurais artificiais, *naive bayes*, máquina de vetores de suporte e *nearest neighbour*. Em todos os casos foram utilizados os valores padrões dos parâmetros dos classificadores, com exceção do algoritmo de redes neurais artificiais que teve seus parâmetros escolhidos a partir de um exemplo de sua documentação, os parâmetros utilizados podem ser vistos na tabela 12.

Parâmetros	Valor
solver	lbfgs
alpha	1e-5
hidden_layer_sizes	(5, 2)
random_state	1

Tabela 12 – Parâmetros utilizados para o classificador de redes neurais artificiais.

7 RESULTADOS

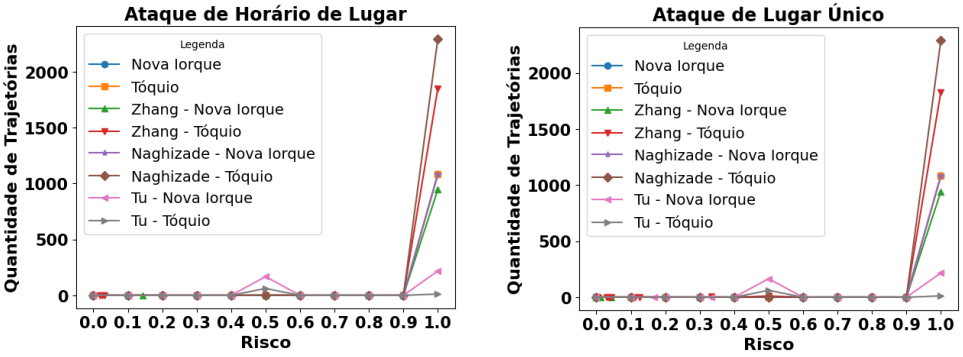
As Figuras 6 a 9 mostram um comparativo do nível de risco encontrado para cada usuário referente aos ataques de lugar e sequência de lugar aplicados aos conjuntos de dados anonimizados e também aos dados originais. Na esquerda do gráfico observa-se a quantidade de trajetórias consideradas pelos ataques, sendo cada trajetória referente a um identificador de usuário único. Na parte inferior do gráfico têm-se o nível de risco de cada trajetória, sendo 0,0 uma proteção máxima e 1,0 uma exposição total do usuário a um atacante com conhecimento prévio.



(a) (b)

Figura 6 – Comparativo do nível de risco dos usuários em referência ao ataque de lugar e ao ataque de casa e trabalho aplicados à todos os dados não anonimizados e anonimizados.

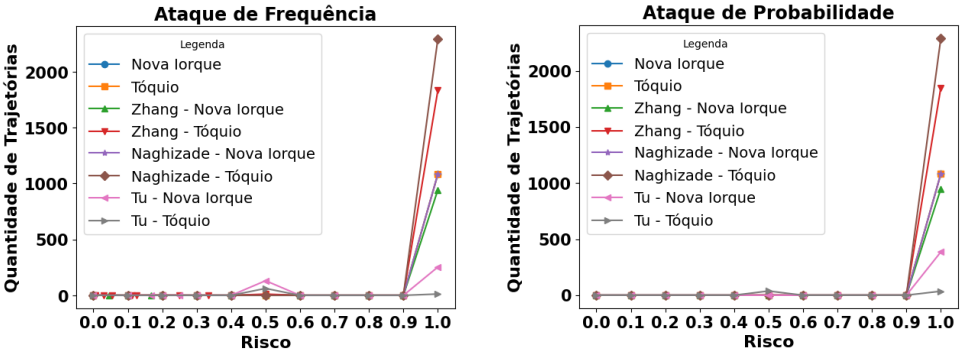
Observa-se que o único ataque que possui alguns usuários sem o nível de exposição máximo nos dados não anonimizados, Nova Iorque e Tóquio, é o ataque de casa e trabalho Figura 6b possuindo para ambos conjuntos de dados um total de 44 usuários com 0,5 de risco de re-identificação.



(a) (b)

Figura 7 – Comparativo do nível de risco dos usuários em referência ao ataque de horário de lugar e ao ataque de lugar único aplicados à todos os dados não anonimizados e anonimizados.

É interessante olhar também em Figura 6b o resultado de (TU et al., 2017) tanto para o conjunto de dados de Tóquio quanto para o de Nova Iorque, Tu - Nova Iorque e Tu - Tóquio, ambos os dados possuem algum ponto que não esteja completamente vulnerável, mas de forma geral possuem mais dados vulneráveis do que antes da anonimização. No entanto, quando se considera a quantidade de pontos anonimizados efetivamente por esse método percebe-se que, para o conjunto de dados de Nova Iorque que teve mais trajetórias anonimizadas em relação ao conjunto de dados de Tóquio, a tendência é ter mais pontos protegidos deste ataque do que os dados não anonimizados. Afinal, quando se compara o risco de exposição dos usuários, especificamente a anonimização desse algoritmo no conjunto de dados de Nova Iorque, aos outros resultados nota-se que esse método é o que teve a maior proporção de usuários protegidos em relação ao total de usuários para os ataques apresentados pelas Figuras 6a, 7a, 7b, 8a, 9a e 9b.



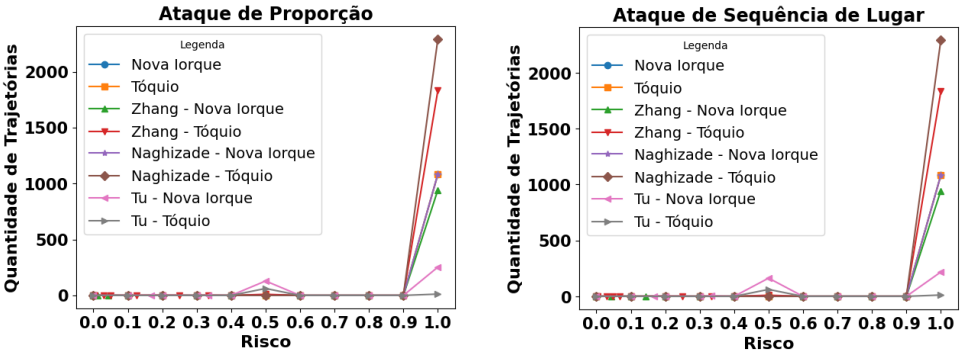
(a) (b)

Figura 8 – Comparativo do nível de risco dos usuários em referência ao ataque de frequência e ao ataque de probabilidade aplicados à todos os dados não anonimizados e anonimizados.

Considerando a relação de usuários que tiveram uma diminuição no seu risco de exposição, os resultados do método de (NAGHIZADE et al., 2020) não se mostraram satisfatórios, pois o único ataque em que há usuários que não possuem risco de exposição 1 é o ataque de casa e trabalho, observado na Figura 6b. Pode-se relacionar esse resultado com o formato dos dados trabalhados, já que o método procura criar segmentos em trajetórias densas similares a coordenadas do sistema de posicionamento global (*GPS*) e não *check-ins*. Durante a anonimização poucas trajetórias foram processadas e efetivamente alteradas, pois muitos pontos não eram categorizados como sensíveis por não terem um ponto *Stop* identificado. Além disso, têm-se que para o conjunto de dados de Nova Iorque a anonimização deixou mais usuários expostos, tendo apenas 12 usuários com um risco de re-identificação de 0,5, 32 a menos se comparado aos dados não anonimizados.

Em relação ao método de (ZHANG et al., 2015), ele não apresenta tantos usuários com um risco de exposição menor quanto (TU et al., 2017), mas nota-se um pequeno aumento na quantidade de usuários que não estão completamente expostos. Em contrapartida, o método de (ZHANG et al., 2015) anonimizou consideravelmente mais rápido e processou muito mais trajetórias do que o método de (TU et al., 2017).

Sobre a Figura 8b, todos os algoritmos demonstraram um desempenho insatisfatório com exceção do resultado do algoritmo de (TU et al., 2017) para o conjunto de dados de Tóquio, que obteve um total de 38 trajetórias com um risco de reidentificação abaixo de 1,0. Contudo, esse resultado também pode ser atribuído a baixa quantidade de trajetórias anonimizadas sobre este conjunto de dados.



(a) (b)

Figura 9 – Comparativo do nível de risco dos usuários em referência ao ataque de proporção e ao ataque de sequência de lugar aplicados à todos os dados não anonimizados e anonimizados.

Nas Tabelas 13 a 18 estão expostos os resultados da acurácia da previsibilidade semântica analisada por algoritmos de aprendizagem de máquina.

O resultado da acurácia para (ZHANG et al., 2015) demonstra bastante o impacto da distribuição das categorias entre os dois conjuntos de dados, como foi apresentado na Figura 4. É possível perceber que no conjunto de dados de Tóquio, o qual possui duas categorias principais: *Business* e *Transport*, obteve-se uma maior anonimização dos dados na questão semântica enquanto nos dados de Nova Iorque, que possui apenas a categoria *Business* com uma quantidade discrepante comparada as outras, a generalização acabou aumentando a acurácia dos algoritmos de aprendizado.

Essa discrepância afeta principalmente esse método, pois ao fazer a troca de

POIs ele não substituí um pelo o outro, apenas pega para si um POI já existente, como a maioria dos pontos desses dados são da categoria *Business* a probabilidade do POI escolhido durante essa troca ser dessa mesma categoria é muito alta. Além disso, os algoritmos de árvores de decisão e de redes neurais artificiais tiveram mais acertos do que a versão não anonimizada.

Métodos	**Anonimizado**	**Não anonimizado**
Árvore de Decisão	0.5566	0.5044
Naive Bayes	0.5258	0.4939
Rede Neural Artificial	0.5258	0.4714
Máquinas de Vetores de Suporte	0.5258	0.4939
Nearest Neighbour	0.4498	0.1910

Tabela 13 – Acurácia do algoritmo de (ZHANG et al., 2015) para o conjunto de dados de Nova Iorque.

Métodos	**Anonimizado**	**Não anonimizado**
Árvore de Decisão	0.4649	0.5613
Naive Bayes	0.4772	0.5617
Rede Neural Artificial	0.4070	0.4880
Máquinas de Vetores de Suporte	0.4767	0.5627
Nearest Neighbour	0.4555	0.5618

Tabela 14 – Acurácia do algoritmo de (ZHANG et al., 2015) para o conjunto de dados de Tóquio.

Quanto aos resultados do método de (NAGHIZADE et al., 2020) tem-se no geral um aumento na exposição das trajetórias, sendo esse aumento muito mais

notável no conjunto de dados de Tóquio. Em relação ao conjunto de dados de Nova Iorque, teve-se dois algoritmos de aprendizado que demonstraram uma acurácia menor. Dado que os pontos considerados para essa análise foram apenas os do tipo *Stop*, e foram encontrados uma pequena quantia desses pontos durante o processo de anonimização, isso tenha aumentado a quantidade de categorias que já eram frequentes nos dados não anonimizados.

Métodos	**Anonimizado**	**Não anonimizado**
Árvore de Decisão	0.5374	0.5044
Naive Bayes	0.4152	0.4939
Rede Neural Artificial	0.4872	0.4714
Máquinas de Vetores de Suporte	0.4661	0.4939
Nearest Neighbour	0.4618	0.1910

Tabela 15 – Acurácia do algoritmo de (NAGHIZADE et al., 2020) para o conjunto de dados de Nova Iorque.

Métodos	**Anonimizado**	**Não anonimizado**
Árvore de Decisão	0.6837	0.5613
Naive Bayes	0.6437	0.5617
Rede Neural Artificial	0.6625	0.4880
Máquinas de Vetores de Suporte	0.6625	0.5627
Nearest Neighbour	0.6640	0.5618

Tabela 16 – Acurácia do algoritmo de (NAGHIZADE et al., 2020) para o conjunto de dados de Tóquio.

Sobre os resultados do método de (TU et al., 2017) não é possível realizar uma análise muito profunda considerado a quantidade de pontos que foi possível

anonimizar, mas é interessante pontuar que de todos os métodos implementados este foi o único que teve uma diminuição na acurácia para o método de aprendizado *nearest neighbour* no conjunto de dados de Nova Iorque, para os dados de Tóquio teve um sutil aumento de 0,03. Para os outros algoritmos de aprendizado a acurácia aumentou consideravelmente.

Métodos	**Anonimizado**	**Não anonimizado**
Árvore de Decisão	0.5488	0.5044
Naive Bayes	0.5245	0.4939
Rede Neural Artificial	0.4705	0.4714
Máquinas de Vetores de Suporte	0.5538	0.4939
Nearest Neighbour	0.1525	0.1910

Tabela 17 – Acurácia do algoritmo de (TU et al., 2017) para o conjunto de dados de Nova Iorque.

Métodos	**Anonimizado**	**Não anonimizado**
Árvore de Decisão	0.7415	0.5613
Naive Bayes	0.7437	0.5617
Rede Neural Artificial	0.6018	0.4880
Máquinas de Vetores de Suporte	0.7519	0.5627
Nearest Neighbour	0.5648	0.5618

Tabela 18 – Acurácia do algoritmo de (TU et al., 2017) para o conjunto de dados de Tóquio.

Por fim, vale constar que o caráter dos dados anonimizados pelos algoritmos de (ZHANG et al., 2015) e de (TU et al., 2017) foram majoritariamente de poucos

pontos por trajetória, tendo em vista que os pontos separados para eles anonimizarem foram a partir do início do conjunto de dados e como apresenta a Figura 5 a maioria das trajetórias no início possuem poucos pontos e muitos usuários.

8 CONCLUSÃO E TRABALHOS FUTUROS

Os resultados obtidos com este trabalho ressaltam que a escolha do método de anonimização é impactada diretamente pelo tipo de dados sendo trabalhado, já que os métodos com base na anonimização de trajetórias densas, ou seja, com muitos pontos que possuem um intervalo de tempo curto entre um e outro não se demonstraram adequados para a anonimização de dados tão esparsos como os de *check-ins*. Uma escolha de método inadequada pode levar a um maior risco de re-identificação dos usuários, como foi o caso do método (NAGHIZADE et al., 2020) tanto para o conjunto de dados de Nova Iorque na questão espaço-temporal quanto para os dados de Tóquio na questão semântica.

Métodos que levam em consideração, durante o processo de anonimização, a preservação da proximidade espaço-temporal dos pontos e regiões construídas entre si, como o (ZHANG et al., 2015) e (TU et al., 2017), obtiveram um menor nível de exposição de indivíduos do conjunto de dados anonimizados quando expostos aos ataques.

Em um comparativo entre os métodos implementados, na questão de preservação da privacidade dos usuários do conjunto de dados utilizado, tem-se que (TU et al., 2017) demonstra ser a escolha mais adequada apesar da demora no tempo de anonimização, tendo uma única queda no desempenho em relação ao ataque de casa e trabalho. O algoritmo (ZHANG et al., 2015), apesar de possuir alguns resultados positivos, eles são ínfimos e percebe-se que a sua estratégia de troca de pontos acaba ficando suscetível a distribuição de categorias semânticas no conjunto de dados, principalmente dado que o resultado de sua anonimização ficou mais previsível para os dados de Nova Iorque que possuem uma expressiva quantidade de pontos da categoria *Business*. Sobre (NAGHIZADE et al., 2020), é interessante notar que apesar da anonimização dos dados gerarem uma maior exposição aos usuários, teve-se duas situações para os dados de Nova Iorque na qual a acurácia da previsibilidade semântica dos dados diminuiu, tendo isso acontecido para os métodos de aprendizado de máquina *naive bayes* e máquinas de vetores de suporte. Esse fato demonstra duas necessidades: a primeira é a utilização de métodos que avaliam o impacto da anonimização tanto na questão de proteção contra a re-identificação de um usuário quanto uma consideração no lado semântico, e a segunda é a variação de métodos

para testar os impactos da anonimização visto que diferentes métodos explicitam o risco em alguns cenários e outros não.

Como trabalhos futuros propõe-se a comparação de métodos voltados a anonimizar um mesmo tipo de dados e a utilização desse tipo de dados na anonimização. É interessante também ter uma análise mais completa na verificação da exposição dos usuários no quesito da semântica da trajetória, além de um aprofundamento no impacto da distribuição semântica dos dados no processo de anonimização.

REFERÊNCIAS

ABUL, Osman; BONCHI, Francesco; NANNI, Mirco. Never walk alone: Uncertainty for anonymity in moving objects databases. In: IEEE. 2008 IEEE 24th international conference on data engineering. [S.l.: s.n.], 2008. p. 376–385.

AKASHKUMAR17. **Classifying data using Support Vector Machines(SVMs) in R**. [S.l.: s.n.]. Disponível em: <https://www.geeksforgeeks.org/classifying-data-using-support-vector-machinessvms-in-r/>.

BRASIL. LEI Nº 13.709, DE 14 DE AGOSTO DE 2018. Dispõe sobre a proteção de dados pessoais e altera a Lei nº 12.965, de 23 de abril de 2014 (Marco Civil da Internet). **Diário Oficial [da] República Federativa do Brasil**, Brasília, DF, 14 ago. 2018. ISSN 1677-7042. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm>. Acesso em: 18 jul. 2022.

BRITO, Felipe T. et al. A Distributed Approach for Privacy Preservation in the Publication of Trajectory Data. In: PROCEEDINGS of the 2nd Workshop on Privacy in Geographic Information Collection and Analysis. Bellevue, WA, USA: Association for Computing Machinery, 2015. (GeoPrivacy'15). ISBN 9781450339698. DOI: 10.1145/2830834.2830835. Disponível em: <https://doi.org/10.1145/2830834.2830835>.

DAI, Yan et al. Personalized Semantic Trajectory Privacy Preservation through Trajectory Reconstruction. **World Wide Web**, Kluwer Academic Publishers, USA, v. 21, n. 4, p. 875–914, jul. 2018. ISSN 1386-145X. DOI: 10.1007/s11280-017-0489-2. Disponível em: <https://doi.org/10.1007/s11280-017-0489-2>.

EUROPEIA, União. General Data Protection Regulation. **Diário Oficial [da] República Federativa do Brasil**, 4 mai. 2016. Disponível em: <https://gdpr-info.eu/>. Acesso em: 18 jul. 2022.

GAVISON, Ruth. Privacy and the Limits of Law. **The Yale Law Journal**, The Yale Law Journal Company, Inc., v. 89, n. 3, p. 421–471, 1980. ISSN 00440094. Disponível em: <http://www.jstor.org/stable/795891>.

GRAMAGLIA, Marco; FIORE, Marco. Hiding Mobile Traffic Fingerprints with GLOVE. In: PROCEEDINGS of the 11th ACM Conference on Emerging Networking Experiments and Technologies. Heidelberg, Germany: Association for Computing Machinery, 2015. (CoNEXT '15). ISBN 9781450334129. DOI: 10.1145/2716281.2836111. Disponível em: <https://doi.org/10.1145/2716281.2836111>.

HUO, Zheng et al. You can walk alone: trajectory privacy-preserving through significant stays protection. In: SPRINGER. INTERNATIONAL conference on database systems for advanced applications. [S.l.: s.n.], 2012. p. 351–366.

LI, Ninghui; LI, Tiancheng; VENKATASUBRAMANIAN, Suresh. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity, p. 106–115, 2007. DOI: 10.1109/ICDE.2007.367856.

MACHANAVAJJHALA, Ashwin et al. L-Diversity: Privacy beyond k-Anonymity. **ACM Trans. Knowl. Discov. Data**, Association for Computing Machinery, New York, NY, USA, v. 1, n. 1, 3–es, mar. 2007. ISSN 1556-4681. DOI: 10.1145/1217299.1217302. Disponível em: <https://doi.org/10.1145/1217299.1217302>.

MITCHELL, Tom M. **Machine Learning**. [S.l.]: AdMcGraw-Hill Science/Engineering/Math, 1997.

NAGHIZADE, Elham et al. Privacy- and Context-Aware Release of Trajectory Data. **ACM Trans. Spatial Algorithms Syst.**, Association for Computing Machinery, New York, NY, USA, v. 6, n. 1, jan. 2020. ISSN 2374-0353. DOI: 10.1145/3363449. Disponível em: <https://doi.org/10.1145/3363449>.

NARKHEDE, Sarang. **Understanding Confusion Matrix**. [S.l.: s.n.]. Disponível em: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>.

NAVIANI, Avinash. **KNN Classification using Scikit-learn**. [S.l.: s.n.]. Disponível em: <https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn>.

NERGIZ, Mehmet Ercan et al. Towards Trajectory Anonymization: A Generalization-Based Approach. **Trans. Data Privacy**, IIIA-CSIC, Bellaterra, Catalonia, ESP, v. 2, n. 1, p. 47–75, abr. 2009. ISSN 1888-5063.

OHM, Paul. Broken promises of privacy: Responding to the surprising failure of anonymization. **UCLA l. Rev.**, HeinOnline, v. 57, p. 1701, 2009.

PAPPALARDO, Luca et al. scikit-mobility: A Python Library for the Analysis, Generation, and Risk Assessment of Mobility Data. **Journal of Statistical Software**, v. 103, n. 1, p. 1–38, 2022. DOI: 10.18637/jss.v103.i04. Disponível em: <https://www.jstatsoft.org/index.php/jss/article/view/v103i04>.

PELLUNGRINI, Roberto et al. A Data Mining Approach to Assess Privacy Risk in Human Mobility Data. **ACM Trans. Intell. Syst. Technol.**, Association for Computing Machinery, New York, NY, USA, v. 9, n. 3, dez. 2017. ISSN 2157-6904. DOI: 10.1145/3106774. Disponível em: <https://doi.org/10.1145/3106774>.

PFITZMANN, Andreas; HANSEN, Marit. Anonymity, unlinkability, unobservability, pseudonymity, and identity management-a consolidated proposal for terminology. Citeseer, 2005.

REIMAN, Jeffrey H. Privacy, Intimacy, and Personhood. **Philosophy & Public Affairs**, Wiley, v. 6, n. 1, p. 26–44, 1976. ISSN 00483915, 10884963. Disponível em: <http://www.jstor.org/stable/2265060>.

RUBENFELD, Jed. The Right of Privacy. **Harvard Law Review**, The Harvard Law Review Association, v. 102, n. 4, p. 737–807, 1989. ISSN 0017811X. Disponível em: <http://www.jstor.org/stable/1341305>.

SALAS, Julián; MEGIAS, David; TORRA, Vicenç. SwapMob: Swapping Trajectories for Mobility Anonymization: UNESCO Chair in Data Privacy, International Conference, PSD 2018, Valencia, Spain, September 26–28, 2018, Proceedings. In: [s.l.: s.n.], jan. 2018. p. 331–346. ISBN 978-3-319-99770-4. DOI: 10.1007/978-3-319-99771-1_22.

SAMARATI, Pierangela; SWEENEY, Latanya. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. technical report, SRI International, 1998.

SOLOVE, Daniel J. Conceptualizing Privacy. **California Law Review**, California Law Review, Inc., v. 90, n. 4, p. 1087–1155, 2002. ISSN 00081221. Disponível em: <http://www.jstor.org/stable/3481326>.

TAN, Pang-Ning et al. **Introduction to Data Mining (2nd Edition)**. 2nd. [S.l.]: Pearson, 2018. ISBN 0133128903.

TU, Zhen et al. Beyond K-Anonymity: Protect Your Trajectory from Semantic Attack. In: p. 1–9. DOI: 10.1109/SAHCN.2017.7964921.

WIKIPÉDIA. **Artificial neural network**. [S.l.: s.n.]. Disponível em: <https://en.wikipedia.org/wiki/Artificial_neural_network>.

_____. **Naive Bayes classifier**. [S.l.: s.n.]. Disponível em: <https://en.wikipedia.org/wiki/Naive_Bayes_classifier>.

YANG, Dingqi et al. Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, v. 45, n. 1, p. 129–142, 2015. DOI: 10.1109/TSMC.2014.2327053.

ZHANG, Zhiqiang et al. An Efficient Method on Trajectory Privacy Preservation. In: p. 231–240. ISBN 978-3-319-22046-8. DOI: 10.1007/978-3-319-22047-5_19.

Anonimização de Trajetórias

Nicole Schmidt, Fernanda O. Gomes, Jean E. Martina

[1]Departamento de Informática e Estatística – Universidade Federal da Santa Catarina (UFSC)
CEP 88040-900 – Florianópolis – SC – Brazil

Abstract. *In the virtual world there are many ways to track different locations where mobile device users pass through, one of the tracking sources are the installed apps on these devices. Publishing this data can expose the privacy of an individual in a harmful way and without consent. This restriction, besides others related to privacy protection, is part of the GDPR [Europeia 2016] and LGPD [Brasil 2018] laws. One option to ensure user privacy is to anonymize the data before its release, ensuring nobody can be linked to it. Some methods used to achieve anonymity are k-anonymity, l-diversity, t-proximity, supression, generalization, and exchange of segments of a trajectory with another. These approaches can be adapted and used together, allowing an analysis of the impact of different methods on data quality and between each other. In order to achieve it, a theoretical review was carried with the algorithms present in the literature and from this review three proposals where chosen. These were implemented together with machine learning powered tests which aim to analyze the results obtained in different categories: anonymization time, data quality and introduced distortion.*

Resumo. *No mundo digital existem formas de rastrear diferentes localizações por onde usuários de dispositivos móveis passam, e uma das fontes desse rastreamento são os próprios aplicativos instalados nesses dispositivos. A publicação desses dados pode expor a privacidade de um indivíduo de maneira danosa e sem o seu consentimento. Essa restrição, além de outras relacionadas a proteção de privacidade, faz parte do recém criado conjunto de leis da GDPR [Europeia 2016] e da LGPD [Brasil 2018]. Uma das ações a serem tomadas para garantir a privacidade dos usuários é anonimizar os dados antes de sua publicação, garantindo que dessa forma o dado não pertencerá a ninguém. Alguns métodos utilizados para alcançar essa anonimidade são a k-anonimidade, l-diversidade, t-proximidade, supressão, generalização e troca de partes de uma trajetória com outra. Essas abordagens podem ser adaptadas e utilizadas em conjunto, permitindo uma análise sobre o impacto que os diferentes métodos tem na qualidade dos dados e entre si. Para isso, fez-se uma revisão teórica dos algoritmos presentes na literatura e dessa revisão se escolheu três propostas alternativas. Elas foram implementadas junto de testes com aprendizado de máquina visando analisar os resultados obtidos em diferentes categorias: o tempo de anonimização, a qualidade dos dados e a distorção introduzida nos dados.*

1. Introdução

No mundo digital existem formas de rastrear diferentes localizações por onde usuários de dispositivos móveis passam, uma das fontes desse rastreamento são os próprios aplicativos instalados nesses dispositivos. Em alguns casos esses dados de localização serão

posteriormente publicados de forma que outras entidades possam analisá-los e utilizá-los para diversos fins, tais como: tomadas de decisão, rotas eficientes, recomendações de localização e mineração de padrões de locomoção.

Deve-se tomar cuidado para que com a publicação desses dados não seja possível identificar um usuário unicamente. A identificação de um indivíduo pode expor a sua privacidade de maneira danosa e sem o seu consentimento. Essa restrição, além de outras relacionadas a proteção de privacidade, fazem parte do recém criado conjunto de leis da GDPR [Europeia 2016] e da LGPD [Brasil 2018], regulamentações para garantir que a privacidade de usuários não seja violada e as quais aplicativos e empresas que lidem com dados de usuários devem obedecer. Uma das ações a serem tomadas para garantir a privacidade dos usuários é anonimizar os dados antes de sua publicação, garantindo que dessa forma o dado não será associado a ninguém.

Um método conhecido e utilizado é o da k-Anonimidade que se baseia em deixar um dado indistinguível de outros k-1 dados. Porém, somente o uso desse método não é suficiente para proteger os dados devido à falta de diversidade que seus resultados proporcionam. Dado essa questão, se faz necessário o uso de outros métodos além da k-anonimidade para alcançar um maior nível de proteção. A diferente combinação de métodos e até a limitação do uso de algumas estratégias como, por exemplo, o uso de supressão, podem ser aplicadas para criar um balanceamento entre a privacidade e a qualidade dos dados sendo processados.

2. Metodologia

Este trabalho foi elaborado a partir de uma pesquisa exploratória com o intuito de trazer uma perspectiva sobre o impacto de diferentes métodos de anonimização da literatura.

Em um primeiro momento, é revisado o estado da arte de anonimização de trajetórias, fazendo também um levantamento dos conceitos básicos de privacidade, anonimização, trajetórias, etc. Em seguida, a partir da revisão do estado da arte foram escolhidos três algoritmos com métodos de anonimização diferentes para implementar.

Na validação dos resultados, a metodologia empregada foi experimental para avaliar o impacto de métodos de anonimização diferentes sobre um mesmo conjunto de dados. Dada a natureza dos resultados envolverem uma análise probabilística tanto para a qualidade quanto para a privacidade dos dados tem-se a utilização de uma abordagem quantitativa nesse aspecto.

Com os algoritmos implementados, compara-se o antes e depois da anonimização de cada algoritmo com uma base de dados. Para essa comparação testa-se a previsibilidade dos dados com o uso de aprendizado de máquina a fim de entender a distorção desse comportamento depois da anonimização e quais outros impactos podem ter acontecido. Também se aplica um conjunto de ataques nos dados anonimizados para analisar o resultado dos métodos de anonimização.

3. Revisão Bibliográfica

3.1. Trajetórias

Uma trajetória é uma sequência de pontos espaço-temporais de um objeto em movimento, em que cada ponto possui a informação sobre o posicionamento do objeto e o tempo em

que o objeto se encontrava naquele lugar. Frequentemente um ponto pi é representado por (xi, yi, ti), sendo xi e yi as coordenadas geográficas em que o ponto se encontra e ti o tempo em que foi registrado o acesso. Se considera que a conexão entre dois pontos é uma linha reta.

3.2. Trajetórias Semânticas

Uma trajetória semântica possui em cada ponto além da informação geoespacial e temporal a informação adicional sobre o lugar em que esse registro ocorreu, como, por exemplo, as coordenadas indicarem que o registro foi feito em uma estação de metrô.

3.3. POIs

Points of Interest (POIs) são pontos da trajetória de um usuário que revelam o valor semântico do lugar e na qual pode-se obter alguma informação do usuário a partir de quanto tempo ele permaneceu nesse determinado ponto de interesse [Zhang et al. 2015]. É possível tratar a informação desses pontos para que ela se torne mais genérica, então ao invés de aparecer que o usuário estava em um restaurante generaliza-se essa informação para dizer que ele estava em um local de comércio.

3.4. ROIs

Regions of Interest (ROIs) são as regiões formadas a partir dos POIs e pode ser representada pelo seu ponto central, raio e tempo de duração [Zhang et al. 2015]. No geral os ROIs ajudam a garantir que aquela região possui uma boa distribuição de POIs.

4. Similaridade

A similaridade entre duas trajetórias é muito considerada quando os algoritmos precisam verificar qual é o custo de unir duas trajetórias para satisfazer a k-anonimidade, quanto de informação vai ser perdida pela generalização de alguns aspectos da trajetória. Esse cálculo pode ocorrer considerando a similaridade ponto a ponto da trajetória [Tu et al. 2017] ou a partir da quantidade de ROIs que as duas trajetórias tem em comum [Zhang et al. 2015].

5. Risco de Exposição

O Risco de exposição de um indivíduo é a sua probabilidade de ser re-identificado em um conjunto de dados móveis. A aplicação de ataques de re-identificação antes da publicação auxiliam a averiguar a melhor troca entre a qualidade dos dados e a privacidade dos usuários a quem esses dados pertencem.

Ataques de re-identificação partem do pressuposto de que o atacante possui um conhecimento prévio acerca de um indivíduo. Os elementos associados ao conhecimento prévio de ataques de re-identificação aplicados a conjunto de dados móveis estão associados a um local e um horário. O risco de re-identificação de um usuário é a mais alta probabilidade de re-identificação das suas instâncias encontrada considerando todas as ocorrências no conjunto de dados.

6. Conjunto de Dados

Os conjuntos de dados utilizados no processo de anonimização são *check-ins* de Nova Iorque e de Tóquio coletados pela *FourSquare* [Yang et al. 2015], com os respectivos tamanhos de 227,428 e 573,703 *check-ins*. Cada registro está associado a um ID de usuário, a uma coordenada com latitude e longitude, fuso horário, um horário UTC e algumas informações semânticas sobre a região em que o registro foi coletado. No contexto deste trabalho se considera que um *check-in* é um ponto de uma trajetória. O conjunto de dados de Tóquio possuí um total de 2293 usuários únicos enquanto Nova Iorque tem 1083.

Cada *check-in* possui uma informação semântica específica associada. Para generalizar as categorias se utilizou de base a separação feita por [Tu et al. 2017] que separa as informações semânticas em seis categorias: entretenimento, educação, cenário, negócios, indústria e residência. No caso deste trabalho se considerou uma categoria adicional: transporte.

7. Algoritmos

7.1. Beyond K-Anonymity: Protect Your Trajectory from Semantic Attack

O algoritmo proposto em [Tu et al. 2017] procura proteger a privacidade dos usuários e manter um nível de qualidade nos dados utilizando k-anonimidade, l-diversidade e t-proximidade. A partir dos dados não tratados generalizam-se os pontos de interesse para 7 categorias: Entretenimento, Educação, Cenário, Comércio, Indústria e Residência. Essa simplificação da informação semântica presente nos POIs é importante, pois enquanto a k-anonimidade é assegurada através da junção de trajetórias, a l-diversidade e a t-proximidade são garantidas na junção de dois pontos de trajetórias diferentes, e é através da diversidade das categorias e da sua distribuição na região em que o ponto se encontra que essas garantias podem ser feitas.

Antes da execução do algoritmo têm a matriz de custo de similaridade entre as trajetórias calculada, o grafo de regiões pronto e cada ponto está associado a alguma categoria generalizada.

Durante a execução as duas trajetórias que possuem entre si o menor custo são escolhidas para serem juntadas. O algoritmo de junção escolhe a trajetória que possui a menor quantidade de pontos para ser juntada na outra trajetória. As regiões semânticas pelas quais as trajetórias passam e o tempo em que estavam nessas regiões é atualizado para cada ponto da nova trajetória gerada.

Esse novo ponto dessa nova trajetória é criado de forma que ele respeite os requisitos da l-diversidade e da t-proximidade, como foi mencionado anteriormente. Caso ele não respeite, procura-se no grafo pelas regiões próximas as já selecionadas e verifica-se quais delas aproximam esse ponto desse critério. Dado as características do conjunto de dados utilizado, caso o ponto sendo juntado à outro não possua nenhum vizinho a ser adicionado o critério não será respeitado com o intuito de evitar um loop infinito.

Após a trajetória resultante ser gerada, as trajetórias que a originaram são excluídas da matriz de custo e, para a nova trajetória, caso ela não respeite o critério da k-anonimidade ainda, terá seu custo calculado com todas as trajetórias restantes na matriz de custo.

7.2. Privacy and Context-aware Release of Trajectory Data

Em [Naghizade et al. 2020] a privacidade é almejada sem que ocorra a junção de uma trajetória com outra. Para isso acontecer se considera que cada ponto que permanece um determinado δt de tempo em uma distância δd é um ponto de parada. Cada ponto de parada possui um indicador do nível de privacidade do local, que permite que caso uma trajetória tenha um ponto de parada com um rigor de privacidade alto ele possa ser alterado para outro ponto de parada realístico que possua um rigor de privacidade menor.

Antes da execução do algoritmo o indicador do nível de privacidade de cada trajetória para cada POI já está configurado e todos os POIs de todas as trajetórias já foram identificados.

Para o algoritmo conseguir imitar pontos de parada mais realísticos nessa troca ele considera as outras categorias identificadas em outros pontos da trajetória e procura alterar esse ponto de parada expositivo para um que seja diferente das categorias já presentes na trajetória e que não seja tão sensitivo.

O ponto de interesse a ser alterado pode ser um presente na própria trajetória, caso contrário será necessário que o algoritmo procure algum POI próximo do que está para ser substituído. A busca utiliza as propriedades de uma elipse para diminuir o tamanho de busca. O primeiro caso é dado como troca de parada, mantendo assim o caminho original da trajetória. O segundo se refere como deslocamento de parada e pode causar alterações tanto temporais quanto espaciais.

O artigo apresenta ainda outra abordagem, mas apenas esta que foi descrita ("Troca Flip-Flop") será considerada neste trabalho. Por uma questão de simplicidade se considerou as mesmas categorias semânticas de [Tu et al. 2017] para os POIs, isso foi necessário para associar o nível de privacidade de um usuário durante a substituição de um POI.

7.3. An Efficient Method on Trajectory Privacy Preservation

A abordagem apresentada em [Zhang et al. 2015] utiliza a k-anonimidade se baseando nos POIs. De forma inicial se extraem os POI existentes nas trajetórias, criam-se as ROIs a partir de *clusters* de POIs e as trajetórias são particionadas com base nas ROIs. Por fim, as trajetórias são anonimizadas em cada conjunto de k-anonimização separado e então publicadas.

Os pontos de interesse possuem uma aproximação diferente de [Tu et al. 2017], nesse caso pontos são considerados POI se: são pontos de parada, pontos de virada, pontos de início ou fim da trajetória. Pontos de virada ocorrem quando o ângulo entre dois segmentos adjacentes passa de um certo limiar definido previamente.

POIs que estão em um tempo-espaço próximo são colocados em um mesmo *cluster*. Para cada ponto de um *cluster* confere-se se o número de vizinhos é menor que um limiar de densidade, e caso positivo esse ponto é marcado como um candidato de ruído preliminar, caso contrário o ponto é considerado um ponto central e ele precisa construir um novo *cluster*.

Depois de todos os pontos no conjunto serem visitados, aqueles marcados como candidatos de ruído são processados. Verifica-se se eles possuem algum vizinho e se

possuírem são adicionados no *cluster* mais próximo, caso contrário serão removidos. Esses clusters são as ROIs. Para alcançar a k-anonimidade é necessário que cada grupo de trajetórias tenha entre k e $2k - 1$ trajetórias. Trajetórias que ao final não foram adicionadas em nenhum grupo são adicionadas em qualquer grupo que tenha pelo menos uma trajetória com uma similaridade maior que um valor arbitrário α. Grupos que possuírem menos que k trajetórias são removidos.

8. Experimentos

O tamanho do conjunto de dados utilizado para cada algoritmo não foi o mesmo devido ao tempo disponível e a demora no processo de anonimização. A quantidade de pontos a serem anonimizados para [Naghizade et al. 2020], [Zhang et al. 2015] e [Tu et al. 2017] em Tóquio foi respectivamente 573.704, 71.712 e 286.852. Em Nova Iorque foi de 227.429, 56.856 e 113.715. Importante notar que apesar da quantidade de dados processada no algoritmo [Tu et al. 2017] se anonimizou no período de 1 semana 754 trajetórias para o conjunto de dados de Nova Iorque e 46 para o de Tóquio.

Os valores utilizados para os parâmetros de [Zhang et al. 2015] foram: ângulo mínimo de 30 graus, distância mínima e raio espacial de 200 metros, tempo de permanência mínimo e raio temporal de 20 minutos, valores de alfa e de *k* sendo respectivamente 0 e 2. Para [Naghizade et al. 2020] o limiar de distância foi 200 metros e o limiar de tempo foi 20 minutos. Os valores de [Tu et al. 2017] para *k*, *l* e *t* foram respectivamente 2, 4 e 0.01.

Para [Naghizade et al. 2020] apenas se considerou trajetórias que possuíssem um mínimo de 2 pontos por dia, em [Tu et al. 2017] o mínimo foi 3. Em relação a [Zhang et al. 2015] não foi exigido um mínimo de pontos pois se realiza uma filtragem de velocidade de trajetória no início de seu pré-processamento.

8.1. Ataques

Para executar os ataques foi necessário processar os dados anonimizados. Cada trajetória construída foi separada em uma lista de pontos com as informações de identificação do usuário inicial, latitude, longitude e marca temporal.

Espera-se que os dados estejam no formato *TrajDataFrame*, para isso é necessário passar a lista de pontos indicando em qual coluna se encontra as informações separadas na etapa anterior. A análise de risco realizada separadamente por cada ataque considera um conhecimento prévio de tamanho 2. A partir da padronização dos dados se chama a rotina de avaliação de risco disponibilizada pela biblioteca *scikit-mobility* [Pappalardo et al. 2022].

8.2. Aprendizado de Máquina

Para avaliar a previsibilidade semântica da categoria de um ponto foi realizado um pré-processamento. Os dados anonimizados de cada algoritmo foram divididos em um conjunto para o treinamento do classificador e outro para o teste, a divisão foi respectivamente 25% e 75%.

Após essa divisão foi construído duas listas relacionadas, sendo a primeira lista com a categoria atual e outra com a próxima categoria. Importante constar que as categorias utilizadas nesse aspecto foram a versão generalizada.

Os algoritmos de aprendizado de máquina utilizados foram: árvores de decisão, redes neurais artificiais, *Naive Nayes*, máquina de vetores de suporte e *nearest neighbour*. Em todos os casos foram utilizados os valores padrões dos parâmetros dos classificadores, com exceção do algoritmo de redes neurais artificiais que teve seus parâmetros escolhidos a partir de um exemplo de sua documentação, os parâmetros utilizados para o resolvedor, alfa, tamanho dos *layers* escondidos e estado aleatório foram respectivamente lbfgs, $1e^{-5}$, (5,2) e 1.

9. Resultados

Observa-se que o único ataque que possui alguns usuários sem o nível de exposição máximo nos dados não anonimizados, Nova Iorque e Tóquio, é o ataque de casa e trabalho possuindo para ambos conjuntos de dados um total de 44 usuários com 0.5 de risco de re-identificação.

Nesse mesmo ataque o resultado de [Tu et al. 2017] tanto para o conjunto de dados de Tóquio quanto para o de Nova Iorque possuem algum ponto que não esteja completamente vulnerável, mas de forma geral possuem mais dados vulneráveis do que antes da anonimização. No entanto, quando se considera a quantidade de pontos anonimizados efetivamente por esse método percebe-se que, para o conjunto de dados de Nova Iorque que teve mais trajetórias anonimizadas em relação ao conjunto de dados de Tóquio, a tendência é ter mais pontos protegidos deste ataque do que os dados não anonimizados. Afinal, quando se compara o risco de exposição dos usuários, especificamente a anonimização desse algoritmo no conjunto de dados de Nova Iorque, aos outros resultados nota-se que esse método é o que teve a maior proporção de usuários protegidos em relação ao total de usuários para o ataque de lugar, ataque de horário de lugar, ataque de lugar único, ataque de frequência, ataque de proporção e ataque de sequência de lugar.

Considerando a relação de usuários que tiveram uma diminuição no seu risco de exposição, os resultados do método de [Naghizade et al. 2020] não se mostraram satisfatórios pois o único ataque em que há usuários que não possuem risco de exposição 1 é o ataque de casa e trabalho. Pode-se relacionar esse resultado com o formato dos dados trabalhados, já que o método procura criar segmentos em trajetórias densas similares a coordenadas do sistema de posicionamento global (*GPS*) e não *check-ins*. Durante a anonimização poucas trajetórias foram processadas e efetivamente alteradas pois muitos pontos não eram categorizados como sensíveis por não terem um ponto *Stop* identificado. Além disso, têm-se que para o conjunto de dados de Nova Iorque a anonimização deixou mais usuários expostos, tendo apenas 12 usuários com um risco de re-identificação de 0.5, 32 a menos se comparado aos dados não anonimizados.

Em relação ao método de [Zhang et al. 2015], ele não apresenta tantos usuários com um risco de exposição menor quanto [Tu et al. 2017], mas nota-se um pequeno aumento na quantidade de usuários que não estão completamente expostos. Em contrapartida o método de [Zhang et al. 2015] anonimizou consideravelmente mais rápido e processou muito mais trajetórias do que o método de [Tu et al. 2017].

Sobre o ataque de probabilidade, todos os algoritmos demonstraram um desempenho insatisfatório com exceção do resultado do algoritmo de [Tu et al. 2017] para o conjunto de dados de Tóquio, que obteve um total de 38 trajetórias com um risco de re-identificação abaixo de 1.0. Contudo esse resultado também pode ser atribuído a baixa

quantidade de trajetórias anonimizadas sobre este conjunto de dados.

Os resultados de acurácia da previsibilidade semântica por algoritmos de aprendizagem de máquina para [Zhang et al. 2015] demonstra bastante o impacto da distribuição das categorias entre os dois conjuntos de dados. Obteve-se uma maior anonimização dos dados na questão semântica para o conjunto de dados de Tóquio, o qual possui duas categorias principais: *Business* e *Transport*, ficando com valores entre 40,70% até 47,72% de acurácia enquanto que nos dados de Nova Iorque, que possui apenas a categoria *Business* com uma quantidade discrepante comparado as outras, a generalização acabou aumentando a acurácia dos algoritmos de aprendizado indo de 44,98% até 55,66%. A acurácia dos dados não anonimizados de Nova Iorque é 19,10% para o método de *Nearest Neighbour* e para os outros varia de 47,14% até 50,44% e para Tóquio varia de 48,80% até 56,27%.

Essa discrepância afeta principalmente esse método pois ao fazer a troca de POIs ele não substituí um pelo o outro, apenas pega para si um POI já existente, como a maior parte de pontos desses dados são da categoria *Business* a probabilidade do POI escolhido durante essa troca ser dessa mesma categoria é muito alta. Além disso, os algoritmos de árvores de decisão e de redes neurais artificiais tiveram mais acertos do que a versão não anonimizada.

Quanto aos resultados do método de [Naghizade et al. 2020] tem-se no geral um aumento na exposição das trajetórias, sendo esse aumento muito mais notável no conjunto de dados de Tóquio que possuem uma acurácia de 64,37% até 68,37%. Em relação ao conjunto de dados de Nova Iorque, teve-se dois algoritmos de aprendizado que demonstraram uma acurácia menor, com os valores de 41,52% para e 46,61% respectivamente para os métodos de *Naive Bayes* e Máquinas de Vetores de Suporte, para os outros métodos a acurácia foi maior que a dos dados não anonimizados. Dado que os pontos considerados para essa análise foram apenas os do tipo *Stop*, e foram encontrados uma pequena quantia desses pontos durante o processo de anonimização, isso tenha aumentado a quantidade de categorias que já eram frequentes nos dados não anonimizados.

Sobre os resultados do método de [Tu et al. 2017] não é possível realizar uma análise muito profunda considerado a quantidade de pontos que foi possível anonimizar, mas é interessante pontuar que de todos os métodos implementados este foi o único que teve uma diminuição na acurácia para o método de aprendizado *Nearest Neighbour* no conjunto de dados de Nova Iorque, para os dados de Tóquio teve um sutil aumento de 0,03. Para os outros algoritmos de aprendizado a acurácia aumentou consideravelmente indo de 60,18% até 75,19% em Tóquio e de 47,05% até 54,88% em Nova Iorque.

Por fim, vale constar que o caráter das dados anonimizadas pelos algoritmos de [Zhang et al. 2015] e de [Tu et al. 2017] foram majoritariamente de poucos pontos por trajetória dado tendo em vista que os pontos separados para eles anonimizarem foram a partir do início do conjunto de dados e a maior parte das trajetórias no início possuem poucos pontos e muitos usuários.

10. Conclusões

Os resultados obtidos com este trabalho ressaltam que a escolha do método de anonimização é impactada diretamente pelo tipo de dados sendo trabalhado, já que os

métodos com base na anonimização de trajetórias densas, ou seja, com muitos pontos que possuem um intervalo de tempo curto entre um e outro não se demonstraram adequados para a anonimização de dados tão esparsos como os de *check-ins*. Uma escolha de método inadequada pode levar a um maior risco de re-identificação dos usuários, como foi o caso do método [Naghizade et al. 2020] tanto para o conjunto de dados de Nova Iorque na questão espaço-temporal quanto para os dados de Tóquio na questão semântica.

Métodos que levam em consideração, durante o processo de anonimização, a preservação da proximidade espaço-temporal dos pontos e regiões construídas entre si, como o [Zhang et al. 2015] e [Tu et al. 2017], obtiveram um menor nível de exposição de indivíduos do conjunto de dados anonimizados quando expostos aos ataques.

Em um comparativo entre os métodos implementados, na questão de preservação da privacidade dos usuários do conjunto de dados utilizado, tem-se que [Tu et al. 2017] demonstra ser a escolha mais adequada apesar da demora no tempo de anonimização, tendo uma única queda no desempenho em relação ao ataque de casa e trabalho. O algoritmo [Zhang et al. 2015], apesar de possuir alguns resultados positivos, eles são ínfimos e percebe-se que a sua estratégia de troca de pontos acaba ficando suscetível a distribuição de categorias semânticas no conjunto de dados, principalmente dado que o resultado de sua anonimização ficou mais previsível para os dados de Nova Iorque que possuem uma expressiva quantidade de pontos da categoria *Business*. Sobre [Naghizade et al. 2020], é interessante notar que apesar da anonimização dos dados gerarem uma maior exposição aos usuários, teve-se duas situações para os dados de Nova Iorque na qual a acurácia da previsibilidade semântica dos dados diminuiu, tendo isso acontecido para os métodos de aprendizado de máquina *naive bayes* e máquinas de vetores de suporte. Esse fato demonstra duas necessidades: a primeira é a utilização de métodos que avaliam o impacto da anonimização tanto na questão de proteção contra a re-identificação de um usuário quanto uma consideração no lado semântico, e a segunda é a variação de métodos para testar os impactos da anonimização visto que diferentes métodos explicitam o risco em alguns cenários e outros não.

Como trabalhos futuros propõe-se a comparação de métodos voltados a anonimizar um mesmo tipo de dados e a utilização desse tipo de dados na anonimização. É interessante também ter uma análise mais completa na verificação da exposição dos usuários no quesito da semântica da trajetória além de um aprofundamento no impacto da distribuição semântica dos dados no processo de anonimização.

References

Brasil (2018). Lei nº 13.709, de 14 de agosto de 2018. dispõe sobre a proteção de dados pessoais e altera a lei nº 12.965, de 23 de abril de 2014 (marco civil da internet). *Diário Oficial [da] República Federativa do Brasil*.

Europeia, U. (2016). General data protection regulation. *Diário Oficial [da] República Federativa do Brasil*.

Naghizade, E., Kulik, L., Tanin, E., and Bailey, J. (2020). Privacy- and context-aware release of trajectory data. *ACM Trans. Spatial Algorithms Syst.*, 6(1).

Pappalardo, L., Simini, F., Barlacchi, G., and Pellungrini, R. (2022). scikit-mobility: A python library for the analysis, generation, and risk assessment of mobility data.

Journal of Statistical Software, 103(1):1–38.

Tu, Z., Zhao, K., Xu, F., Li, Y., Su, L., and Jin, D. (2017). Beyond k-anonymity: Protect your trajectory from semantic attack. pages 1–9.

Yang, D., Zhang, D., Zheng, V. W., and Yu, Z. (2015). Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):129–142.

Zhang, Z., Sun, Y., Xie, X., and Pan, H. (2015). An efficient method on trajectory privacy preservation. pages 231–240.