

Uma Implementação Distribuída em Névoa do Algoritmo de Detecção de Novidade em Fluxos de Dados MINAS

Luís Henrique Puhl de Souza

Orientador: Prof. Dr. Hermes Senger

Fevereiro 2020

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

Departamento de Computação

Programa de Pós-Graduação em Ciência da Computação

1. Introdução
2. Fundamentos
3. Estado da Arte e Trabalhos Relacionados
4. Proposta
5. Resultados Preliminares
6. Considerações Finais

Introdução

Crescimento do número de dispositivos IoT e riscos associados;

Deteção de intrusão em redes por novidade

Um sistema para deteção de intrusão em Redes IoT implementando em névoa

A hipótese do trabalho é que o algoritmo MINAS pode ser distribuído em nós de nuvem e névoa reduzindo a latência e com pouco comprometimento na qualidade de deteção.

Nota: pode falar:

- *difficuldade de atualização de SW*
- *como foi o ataque MIRAI? (só curiosidade)*
- *capacidade de autodefesa? mesmo outros computadores não têm...*

Fundamentos

Ambientes de computação Distribuída;

Plataformas de processamento distribuído de fluxos;

Métodos Detecção de Novidade;

Nota: Por que "seria difícil processar de forma centralizado"?

Definição de Fluxo de Dados

Defu

Evolução;

Mudança;

Ruído;

Métodos Detecção de Novidade

Métodos Detecção de Novidade (*Novelty Detection*) lidam com o reconhecimento e classificação de exemplos em padrões que diferem de padrões anteriores (PERNER, 2007; GAMA; RODRIGUES, 2010).

Conforme Gama e Rodrigues (2010), são características de fluxos de dados contínuos:

- Evolução de conceito (*Concept Evolution*);

- Mudança de conceito (*Concept Drift*, deriva ou desvio);

- Ruído e *outliers*;

Nota: Poderia explicar melhor a evolução de modelos, "concept drift" (precisa estudar um pouco + a teoria).

Ambientes de computação Distribuída

Computação em Nuvem (*Cloud Computing*):

Características: Serviço sob Demanda, Amplo acesso à rede, Agrupamento de recursos, Elasticidade, Serviço mensurado;

Implementações: Nuvem privada, Nuvem comunitária, Nuvem pública, Nuvem híbrida (MELL; GRANCE, 2012).

Nota: Separar os itens em bullets (sub-bullets)

Ambientes de computação Distribuída

Computação de Borda (*Edge Computing*) (SHI et al., 2016):

Refere-se a qualquer recurso computacional ou de rede entre os dispositivos de borda e centro de dados hospedados em nuvem.

Computação em Névoa (*Fog Computing*) (BONOMI et al., 2012; DASTJERDI; BUYYA, 2016):

Características: Mobilidade, Heterogeneidade, Baixa Latência, Distribuição geográfica, Alto número de nós, Interoperabilidade e federação, Uso de fluxo de dados e aplicações em tempo real (IEEE Communications Society, 2018).

Nota: Definição de Fog e borda...

São intercambiáveis mesmo?

*(*Evite colocar questões inconclusivas)*

Plataformas de processamento distribuído de fluxos

Mineração de Dados e Fluxo de Dados;

Arquiteturas *Lambda* e *Kappa*;

MapReduce e *Apache Hadoop*;

Apache Spark, *Resilient Distributed Dataset* e *micro-batching* para *Spark Streaming*;

Apache Storm;

Apache Flink;

Nota: Faltou definir antes o que é processamento de fluxo.

fluxo vs lote deixar mais claro a diferença

Nota: O slide fala de plataformas, mas vc falou de kappa e lambda que não deveria falar neste slide

Nota: Muita fala p/ esse slide (ficou explicando cada fundamento). Deveria quebrar o slide em 3 ou 4 slides ao menos.

Fundamentos

```
DataStream<String> lines = env.addSource(  
    new FlinkKafkaConsumer<> (...));  
  
DataStream<Event> events = lines.map((line) -> parse(line));  
  
DataStream<Statistics> stats = events  
    .keyBy("id")  
    .timeWindow(Time.seconds(10))  
    .apply(new MyWindowAggregationFunction());  
  
stats.addSink(new BucketingSink(path));
```

Source

Transformation

Transformation

Sink

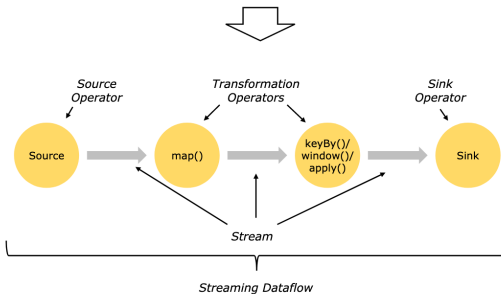


Figura 1: Exemplo de código e *data flow* do Apache Flink (Apache Flink, 2020)

Algoritmo MINAS

Algoritmo e suas estratégias:

Modelo de aprendizado *Offline-Online*;

Transformação dos dados analisados para o espaço \mathbb{R}^d ;

Função de classificação baseada em distância euclideana;

Modelo de classificação com *Clusters*;

Algoritmo de agrupamento para identificação de padrões;

Nota: MINAS: Use figura!
pseudocodigo?

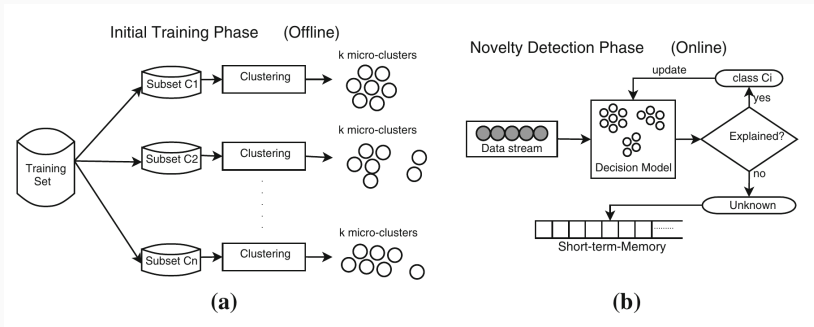


Figura 2: Visão geral do algoritmo MINAS com fases *Offline* (a) e *Online* (b) (FARIA; CARVALHO; GAMA, 2015)

Nota: Explique o desenho! o que significa cada objeto da figura? explique na figura.

Estado da Arte e Trabalhos Relacionados

Extensões do Algoritmo MINAS;

Sistemas de detecção de intrusão em redes;

*Nota: Trabalhos relacionados * extensões? esses são seus concorrentes?
qual é a relação?*

MEIO FORA DE FOCO!

Extensões do Algoritmo MINAS

FuzzyND: extensão do algoritmo original para classificação com conjunto de etiquetas *fuzzy* (Da Silva et al., 2018; SILVA, 2018);

MINAS-LC e MINAS-BR: extensão do algoritmo original tratando classificação multi-etiquetas (COSTA et al., 2019; COSTA, 2019);

Sistemas de detecção de intrusão em redes

Ferramenta BigFlow (VIEGAS et al., 2019);

Ferramenta CATRACA (LOPEZ, 2018);

Arquitetura IDSA-IoT (CASSALES et al., 2019);

Nota: OK, isso é trabalho relacionado!

*Dissecar melhor [bigflow, catraca, idsa-iot]
passar mais rápido, detalhar nos próximos.*

Estado da Arte e Trabalhos Relacionados

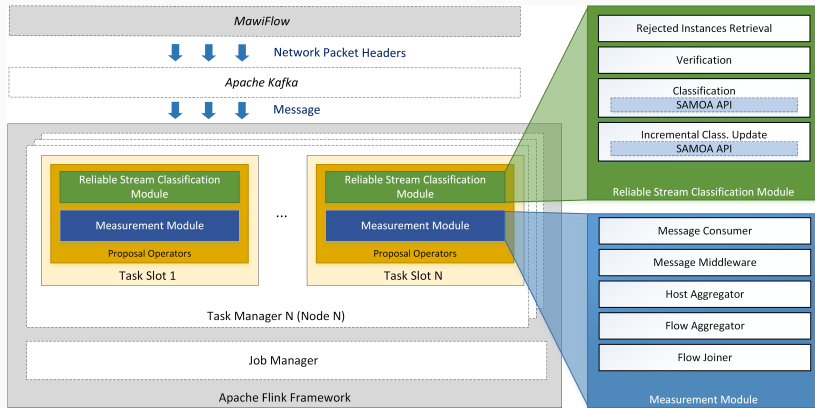


Figura 3: Visão geral da arquitetura e distribuição da ferramenta BigFlow (VIEGAS et al., 2019).

*Nota: BigFlow usa stream mining?
- qual é a grande contribuição?*

Estado da Arte e Trabalhos Relacionados

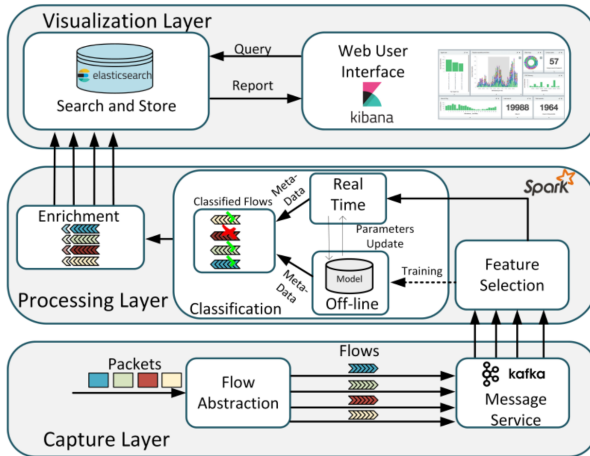


Figura 4: Arquitetura em camadas da ferramenta CATRACA (LOPEZ, 2018).

Estado da Arte e Trabalhos Relacionados

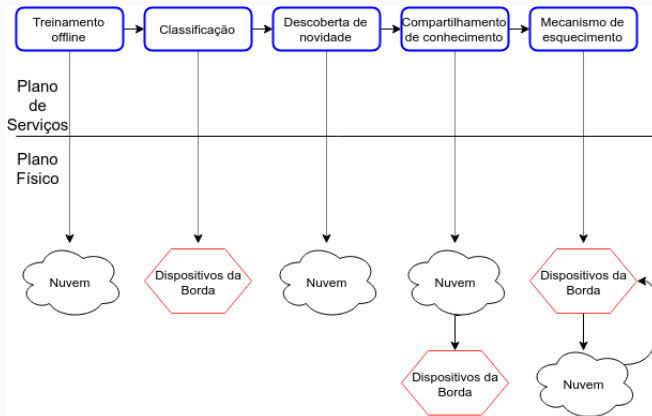


Figura 5: Distribuição de Serviços da Arquitetura IDSA-IoT. Produzida e traduzida por Cassales et al. (2019).

Proposta

Plataforma de processamento distribuído;

Arquitetura IDS-IoT;

Distribuição do algoritmo MINAS;

Validação da implementação por comparação de métricas de qualidade;

Métricas de escalabilidade;

Nota: Proposta

*- poderia fazer algumas "perguntas" antes de fazer a sua proposta
⇒ quais perguntas? (voce precisa saber quais)*

O sistema M-FOG é dividido em 5 módulos subdivididos em 2 grupos.

Módulos principais implementam o algoritmo MINAS

módulo treinamento (*Training Module*);

módulo classificador (*Classification Module*);

módulo detector de novidades (*Novelty Detection Module*).

Módulos auxiliares, utilizados para avaliação

módulo auxiliar *source* (fonte);

módulo auxiliar *sink* (sorvedouro, consumidor final).

Proposta

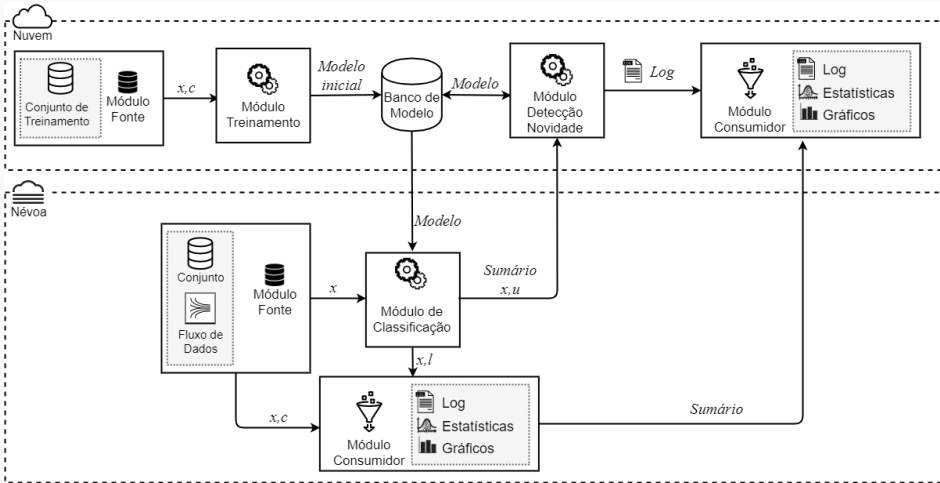


Figura 6: Arquitetura e fluxos de dados do sistema M-FOG.

Resultados Preliminares

Primeira Implementação com Python e Apache Kafka

Python é acessível e fornece bibliotecas diversas;

Apache Kafka é um sistema de mensagens distribuído;

Interface de programação com cliente produtor e consumidor;

Mensagens organizadas em tópicos que são distribuídos em partições;

A hipótese de que a carga seria distribuída entre os consumidores, uma vez que o consumidor pode selecionar uma partição para leitura;

Em experimento com um produtor, 8 partições e 8 consumidores, observou-se que um consumidor processava a maior parte das mensagens, poucos consumidores recebiam algumas mensagens e a maioria dos consumidores não recebia mensagem alguma.

Nota: CUIDADO, o fato de vc não ter conseguido não significa que não é possível

concluir que "o sistema não escala ..." (pode citar, só tome cuidado com a conclusão tirada)

Segunda Implementação com Apache Flink

Implementação escrita em Scala ou Java;

Processamento de fluxos *Stateful*;

Falta de bibliotecas que distribuam algoritmos base como *K-means*;

Sistema *M-FOG* em desenvolvimento, atualmente na fase de validação através das métricas de qualidade de classificação.

Nota: métricas: Tem a fórmula? expressão?

Nota: "Avaliação do fluxo de saída do classificador" isto é uma métrica?

Nota: Incluir CR

Métricas e Ambientes

Métricas de qualidade de classificação:

- Avaliação do fluxo de saída do classificador;

- Uso de uma matriz de confusão ou erro;

- Taxa de desconhecidos;

- Macro F-score;

Métricas de escalabilidade:

- Número e tipo de processadores;

- Uso de memória;

- Tempo de processamento;

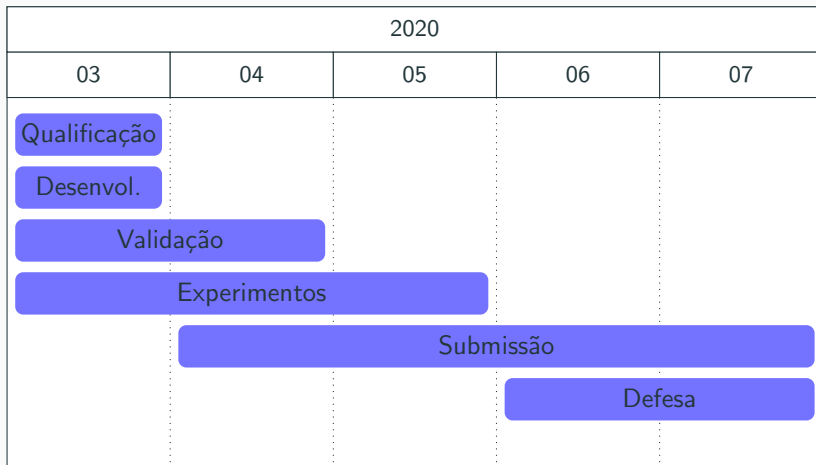
- Taxa de eventos;

- Latência entre a produção e classificação.

Considerações Finais


Trabalho continua com a finalização da implementação e validação do MFOG com MINAS.


Cronograma





Obrigado!


 Apache Flink. *Apache Flink*. 2020. Disponível em: <https://flink.apache.org/>.


 BONOMI, F. et al. Fog computing and its role in the internet of things. In: *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. [s.n.], 2012. p. 13–16. ISBN 9781450315197. Disponível em: <http://www.lispmob.org>.


 CASSALES, G. W. et al. IDSA-IoT: An Intrusion Detection System Architecture for IoT Networks. In: *2019 IEEE Symposium on Computers and Communications (ISCC)*. [s.n.], 2019. p. 1–7. ISBN 978-1-7281-2999-0. ISSN 1530-1346. Disponível em: <https://ieeexplore.ieee.org/document/8969609/>.


 COSTA, J. D. *Deteção De Novidade Em Fluxos Contínuos De Dados Multirrótulo*. 127 p. Tese (Master) — UNIVERSIDADE FEDERAL DE SÃO CARLOS, 2019. Disponível em: <https://repositorio.ufscar.br/handle/ufscar/12197>.


 COSTA, J. D. et al. Novelty detection for multi-label stream classification. *Proceedings - 2019 Brazilian Conference on Intelligent Systems, BRACIS 2019*, n. 8, p. 144–149, 2019.


 Da Silva, T. P. et al. A fuzzy multiclass novelty detector for data streams. *IEEE International Conference on Fuzzy Systems*, IEEE, v. 2018-July, p. 1–8, 2018. ISSN 10987584.


 DASTJERDI, A. V.; BUYYA, R. Fog computing: Helping the internet of things realize its potential. *Computer*, IEEE, v. 49, n. 8, p. 112–116, Aug 2016. ISSN 1558-0814.


 FARIA, E. R. d.; CARVALHO, A. C. Ponce de L. F.; GAMA, J. Minas: multiclass learning algorithm for novelty detection in data streams. *Data Mining and Knowledge Discovery*, v. 30, n. 3, p. 640–680, May 2015. ISSN 1573-756X. Disponível em: <https://doi.org/10.1007/s10618-015-0433-y>.


 GAMA, J.; RODRIGUES, P. P. *Knowledge Discovery from Data Streams*. [S.l.]: Chapman and Hall/CRC, 2010. ISBN 9781439826119.


 IEEE Communications Society. *IEEE Std 1934-2018: IEEE Standard for Adoption of OpenFog Reference Architecture for Fog Computing*. IEEE, 2018. 176 p. ISBN 9781504450171. Disponível em: [⟨https://ieeexplore.ieee.org/document/8423800⟩](https://ieeexplore.ieee.org/document/8423800).


 LOPEZ, M. E. A. *A monitoring and threat detection system using stream processing as a virtual function for Big Data*. Tese (Theses) — Sorbonne Université ; Universidade federal do Rio de Janeiro, Jun 2018. Disponível em: [⟨https://tel.archives-ouvertes.fr/tel-02111017⟩](https://tel.archives-ouvertes.fr/tel-02111017).

 MELL, P.; GRANCE, T. The NIST definition of cloud computing: Recommendations of the National Institute of Standards and Technology. In: NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY. *Public Cloud Computing: Security and Privacy Guidelines*. 2012. p. 97–101. ISBN 9781620819821. Disponível em: [⟨http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf⟩](http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf).

 PERNER, P. Concepts for novelty detection and handling based on a case-based reasoning process scheme. In: *Advances in Data Mining. Theoretical Aspects and Applications*. [S.l.]: Springer, 2007. p. 21–33. ISBN 978-3-540-73435-2.

 SHI, W. et al. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, Institute of Electrical and Electronics Engineers Inc., v. 3, n. 5, p. 637–646, oct 2016. ISSN 23274662. Disponível em: [〈https://ieeexplore.ieee.org/abstract/document/7488250〉](https://ieeexplore.ieee.org/abstract/document/7488250).

 SILVA, T. P. da. *Abordagem Fuzzy para Detecção de Novidade em Fluxo Contínuo de Dados*. 89 p. Tese (Master) — Universidade Federal de São Carlos, 2018. Disponível em: [〈https://repositorio.ufscar.br/handle/ufscar/10544〉](https://repositorio.ufscar.br/handle/ufscar/10544).

 VIEGAS, E. et al. Bigflow: Real-time and reliable anomaly-based intrusion detection for high-speed networks. *Future Generation Computer Systems*, v. 93, p. 473 – 485, 2019. ISSN 0167-739X. Disponível em: [〈http://www.sciencedirect.com/science/article/pii/S0167739X18307635〉](http://www.sciencedirect.com/science/article/pii/S0167739X18307635).

Recomendações de Leitura

empty