UNIVERSIDADE FEDERAL DE PELOTAS
Centro de Desenvolvimento Tecnológico
Programa de Pós-Graduação em Computação

Dissertação

Aprendizado de Máquina Aplicado à Previsão de Infestações de Pragas Através de Mudanças Meteorológicas

William Dalmorra de Souza

Pelotas, 2019

William Dalmorra de Souza

Aprendizado de Máquina Aplicado à Previsão de Infestações de Pragas Através de Mudanças Meteorológicas

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Mestre em Ciência da Computação

Orientador: Prof. Dr. Paulo Roberto Ferreira Júnior
Coorientador: Prof. Dr. Marilton Sanchotene de Aguiar

Pelotas, 2019

Universidade Federal de Pelotas / Sistema de Bibliotecas
Catalogação na Publicação

S719a Souza, William Dalmorra de

Aprendizado de máquina aplicado à previsão de infestações de pragas através de mudanças meteorológicas / William Dalmorra de Souza ; Paulo Roberto Ferreira Júnior, orientador ; Marilton Sanchotene de Aguiar, coorientador. — Pelotas, 2019.

43 f.

Dissertação (Mestrado) — Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, 2019.

1. Aprendizado de máquina. 2. Aprendizado online. 3. Controle de infestação. 4. Dados meteorológicos. I. Ferreira Júnior, Paulo Roberto, orient. II. Aguiar, Marilton Sanchotene de, coorient. III. Título.

CDD : 005

Elaborada por Aline Herbstrith Batista CRB: 10/1737

RESUMO

SOUZA, William Dalmorra de. **Aprendizado de Máquina Aplicado à Previsão de Infestações de Pragas Através de Mudanças Meteorológicas**. 2019. 43 f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2019.

O uso de pesticidas como forma de controle do avanço da população de uma determinada praga é a técnica mais utilizada atualmente. Uma das razões para este fato é o tempo de resposta que tal método possui, agindo de forma rápida e eliminando a ameaça à plantação. Entretanto, tais pesticidas são conhecidos por seus riscos à saúde, tanto dos consumidores dos produtos quanto dos trabalhadores rurais e seus aplicadores. As infestações são causadas por insetos, e muitos desses insetos possuem características que são fortemente influenciadas por fatores meteorológicos, por exemplo, o fato de serem ectotérmicos, o que os tornam frágeis à alterações na temperatura da região. Com base neste conhecimento, é possível afirmar que o comportamento dos insetos é previsível, capaz de ser determinado a partir das alterações climáticas da região. Neste contexto, este trabalho propõe a criação de um modelo de previsão de infestações baseando-se nas alterações climáticas da região. Mais especificamente, este trabalho tem como objetivo ser capaz de informar a população dos insetos presentes nas plantações com uma antecedência de uma semana, baseando-se nos dados meteorológicos da semana anterior. As informações meteorológicas utilizadas para a previsão foram a média semanal da temperatura e do nível de chuva. Além disso, foi utilizada a informação populacional corrente na plantação como entrada para o modelo. Para a criação e validação do modelo, foi criado um *toy dataset* a partir de uma rede LSTM treinada com dados do Instituto Biológico de Campinas. Para a previsão foi criada uma rede LSTM com aprendizado online. Para comparação, foi treinada outra rede LSTM com os mesmos dados, porém ela foi dividida em um conjunto de treinamento e um conjunto de teste. Ao final dos experimentos, pode-se verificar que o modelo online obteve um Erro Médio Quadrático inferior ao modelo tradicional.

Palavras-Chave: aprendizado-de-máquina; aprendizado-online; controle-de-infestação; dados-meteorológicos

ABSTRACT

SOUZA, William Dalmorra de. **Machine Learning Applied to Predicting Pest Infestations Through Weather Changes**. 2019. 43 f. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2019.

The use of pesticides for controlling the population growth of a given pest is the most commonly used technique. One of the reasons for this is the response time that such method has, acting quickly and eliminating the threat to the crops. However, such pesticides are known for their health risk to the consumers of the products, the rural workers, and the pesticide appliers. As infestations are caused by insects, and many of these insects have characteristics that are strongly influenced by weather factors, for example, being ectothermic, which makes them fragile to temperature changes in the region. Based on this, it is possible to state that the behavior of insects is predictable, capable of being determined by the climatic changes in the region. In this context, this work proposes the creation of a model for prediction of an infestation based on the climatic changes in the region. More specifically, this work aims at predicting the insect population in a crop one week in advance. The meteorological data used in predictions were the temperature and rain weekly average. Besides that, the insect population information current in the crop was also used as an input for the model. To create and validate the model, a toy dataset was created from a LSTM network with data from Campinas Biological Institute. For the prediction, a new online learning LSTM network was designed. However, it was split into a training dataset and a test dataset. At the end of experiments, it could be attested that the online model achieved a Mean Squared Error less than the tradicional model.

Keywords: machine-learning; online-learning; pest-control; meteorological-data

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

MIP	Manejo Integrado de Pragas
ML	*Machine Learning*
RNN	*Recurrent Neural Network*
LSTM	*Long Short Term Memory*
MSE	*Mean Squared Error*

SUMÁRIO

1 INTRODUÇÃO

A agricultura é a principal fonte econômica para muitos países, tais como, Estados Unidos, Canadá, França e Brasil. Para muitos destes, a agricultura é responsável por 7-10% da economia total do país (BOLSO, 2014). Os produtos agrícolas são utilizados de diversas formas no cotidiano da humanidade, seja para alimentação própria como para a alimentação de outros animais criados exclusivamente para consumo. Além disso, a população mundial segue crescendo a cada ano, a uma taxa de 75 milhões de pessoas aproximadamente. Este aumento na população gera também um aumento na demanda pelos produtos agrícolas (TROSTLE et al., 2008). Porém, a prática da agricultura possui diversas adversidades que dificultam a vida dos agricultores nos dias atuais, como, por exemplo, a infestação de pragas nas lavouras.

Existem diversos animais que são considerados danosos às plantações de vegetais e frutas. Estas pragas são classificadas de acordo com a parte do produto cultivado que é afetada por elas, como as folhas, o fruto, o caule ou a raiz. Tais pragas podem ser devastadoras em uma lavoura, causando prejuízos enormes e muitas vezes irreversíveis para os responsáveis se não forem controladas em tempo hábil. Por exemplo, os percevejos nas lavouras de soja são capazes de reduzir a qualidade, vigor e viabilidade das sementes, além das mesmas serem encontradas com alterações nos teores de proteína e óleo (CORRÊA-FERREIRA; PANIZZI, 1999). Como forma de proteção contra tais insetos, os produtores buscam auxílio na utilização de agrotóxicos capazes de controlar as pragas. Estes agrotóxicos porém, acabam por afetar os produtos se não forem utilizados corretamente (ZANATTA et al., 2007) e geram um gasto financeiro alto para o produtor. Além disso, estes produtos são, muitas vezes, aplicados de forma negligente, no intuito de prevenção, em momentos que não seria necessário, o que causa um aumento de gastos que poderia ser evitado.

Estes agrotóxicos, além disso, podem causar danos à saúde, tanto dos consumidores quanto dos responsáveis pela aplicação do produto. Existem diversos estudos relacionados ao uso de agrotóxicos e enfermidades associadas. Por exemplo, em MOSTAFALOU; ABDOLLAHI (2013), são listadas diversas doenças crônicas e as evidências da relação de seu desenvolvimento devido à exposição a agrotóxicos. Al-

gumas das doenças listadas são: câncer, malformações congênitas, distúrbios reprodutivos, além de danos genéticos que podem ocorrer, como é mostrado em KHAYAT et al. (2013). Neste trabalho, foram selecionados 72 trabalhadores rurais e divididos em 2 grupos, um grupo era composto por trabalhadores que sofreram exposição a agrotóxicos durante sua carreira e outro grupo onde os trabalhadores nunca tiveram contato com tal produto. Todos trabalhadores foram testados e, ao final do estudo, os autores comprovaram que o grupo que trabalhava com os agrotóxicos apresentavam um maior dano genômico.

Como insetos são ectotérmicos, qualquer variação de temperatura na região afeta fortemente a temperatura corporal desses animais, logo eles são diretamente influenciados pelas alterações climáticas do local onde estão. Além da temperatura, umidade, precipitação, velocidade do vento, entre outras, são variáveis também conhecidas por influenciar a vida dos insetos (PORTER; PARRY; CARTER, 1991). As mudanças no clima podem afetar desde a distribuição geográfica das espécies e a probabilidade de invasão por pestes migrantes, quanto o número de gerações produzidas pelos insetos e a taxa de crescimento da população (KOCMÁNKOVÁ et al., 2009). Atualmente existem trabalhos que utilizam variações meteorológicas para prever o comportamento de inimigos naturais das pragas (THOMSON; MACFADYEN; HOFFMANN, 2010). Outros trabalhos mostram como as mudanças climáticas afetam a migração de algumas pragas (CANNON, 1998) ou os efeitos que o clima tem sobre os insetos nas plantações (PORTER; PARRY; CARTER, 1991).

Para auxiliar no controle do uso de agrotóxicos, é utilizado um conjunto de técnicas conhecido como Manejo Integrado de Pragas (MIP). Tais técnicas buscam reduzir a contaminação nos alimentos plantados, seja pelo uso de meios alternativos como a aplicação de inimigos naturais das pragas nas lavouras, (controle biológico) ou a queima de parte da plantação, evitando que a infestação atinja partes ainda saudáveis. Outra técnica utilizada é a distribuição de armadilhas com algum tipo de feromônio (sexual, alimentício, etc...) que visa atrair os insetos para que haja um controle do crescimento populacional dos insetos. Esta técnica permite que, ao perceber um aumento considerado perigoso do número de insetos, seja aplicado agrotóxico suficiente para controlar o avanço da população. Desta forma, é possível evitar danos que causariam prejuízos à produção e ao mesmo tempo, aplicar os produtos somente quando for realmente necessário. Entretanto, este conjunto de técnicas não é capaz de automaticamente prever uma infestação antes que ela ocorra, somente disponibilizam informações para facilitar a identificação de uma contaminação ocorrente.

Diante disto, este trabalho propõe o desenvolvimento de um modelo de previsão de infestações de pragas nas lavouras utilizando Redes Neurais Recorrentes (do inglês, *Recurrent Neural Networks - RNN*). Para a previsão, o modelo baseia-se nas alterações meteorológicas na região e o acompanhamento da população dos inse-

tos. O modelo deve ser capaz de aprender *online*, ou seja, seja retreinado com os novos dados que são apresentados ao longo do tempo. Isto permite uma adaptação automática do modelo para diferentes espécies de insetos em diferente regiões com características climáticas diferentes. A base de dados para treinamento do modelo é alimentada por dados recebidos de estações meteorológicas e do acompanhamento do número de insetos presentes nas armadilhas em campo, que deve ser obtido através de armadilhas inteligentes capazes de identificar e contar os insetos capturados e enviar em tempo real para a base de dados. Tais armadilhas inteligentes estão sendo desenvolvidas paralelamente à este projeto. O uso de uma RNN baseia-se na necessidade de aprender com os resultados dos exemplos passados, aprendendo assim uma conexão temporal entre os dados de entrada, uma vez que tal informação é relevante no domínio da aplicação.

Com este modelo de predição, pretende-se então causar a redução no uso de agrotóxicos nas plantações ao ter a possibilidade de controlar uma infestação que está ainda em desenvolvimento. O número de insetos reduzido requer uma menor concentração de produtos para que a infestação não atinja um nível de dano econômico considerado alto para os produtores.

Por fim, destaca-se também que este trabalho é parte de um projeto maior que também possui duas outras frentes, sendo elas: a automatização da contagem e identificação dos insetos capturados nas armadilhas a partir de uma câmera e uma placa Raspberry Pi acopladas às armadilhas; e a comunicação entre as armadilhas presentes no campo através de uma rede de sensores sem fio para transmissão das informações coletadas. Este projeto está sendo desenvolvido desde 2016 e já foram publicados dois artigos em eventos internacionais. O primeiro foi apresentado na *Mexican International Conference on Artificial Intelligence* com qualis B1 em 2017. O segundo foi apresentado no *Symposium on Applied Computing* com qualis A1 em 2018. Além disso, o projeto foi premiado no *Google Research Awards for Latin America* em 2018, sendo novamente premiado em 2019.

Nos próximos capítulos serão apresentados um estudo sobre Redes Neurais Recorrentes e suas aplicações (2), a metodologia aplicada no treinamento e modelagem da rede neural utilizada para desenvolver o modelo (4), e por fim algumas conclusões finais sobre o projeto e trabalhos futuros para serem desenvolvidos (5).

2 REDES NEURAIS RECORRENTES

2.1 Aprendizado de Máquina

Aprendizado de Máquina (do inglês, *Machine Learning - ML*) é uma sub-área da Inteligência Artificial. ML diz que quando o desenvolvedor não possui conhecimento total sobre o domínio da aplicação, é impossível a criação manual de um modelo capaz de realizar a tarefa requisitada. Aprendizado de Máquina pode ser considerado a única maneira de possibilitar o modelo adquirir sozinho o conhecimento necessário para a execução desta tarefa. Logo, pode-se dizer que ML provê autonomia para o modelo (RUSSELL et al., 2003).

Como parte da Inteligência Artificial, ML tem sido a primeira escolha para o desenvolvimento de aplicações como reconhecimento de voz, visão computacional, processamento de linguagens naturais, entre várias outras aplicações. Isso se deve ao fato de diversos desenvolvedores afirmarem que é muito mais simples treinar um sistema mostrando as entradas e quais são as saídas apropriadas, como faz muitas técnicas de ML, do que programar manualmente qual a saída desejada para cada caso possível (JORDAN; MITCHELL, 2015). Existem dois tipos principais de ML que podem ser classificados como: aprendizado não-supervisionado e supervisionado. Existe ainda o conceito de aprendizado semi-supervisionado, que busca combinar os dois conceitos anteriores (WITTEN; FRANK, 2005).

2.1.1 Aprendizado Não-Supervisionado

O aprendizado não-supervisionado recebe um conjunto de dados chamado Conjunto de Treinamento que contém apenas o vetor de atributos da amostra. A saída é totalmente desconhecida. Logo, o algoritmo deve classificar o melhor possível os dados, separando-os em semi-conjuntos, a partir de suas semelhanças nos atributos. O algoritmo de clusterização é o principal exemplo de aprendizado não-supervisionado (MONARD; BARANAUSKAS, 2003).

2.1.2 Aprendizado Supervisionado

O aprendizado supervisionado é o tipo de aprendizado que busca encontrar um padrão entre os conjuntos de atributos de suas amostras de entrada com suas respectivas saídas. Logo, é fornecido ao algoritmo, assim como no aprendizado não-supervisionado, um Conjunto de Treinamento que contém normalmente um vetor de atributos para cada amostra, porém desta vez, o conjunto deve conter também qual a saída esperada para cada entrada. O objetivo é descobrir a relação dos atributos com a saída para que possa então, em novas amostras fornecidas futuramente, descobrir a saída correta. O aprendizado supervisionado é utilizado no desenvolvimento de algoritmos de classificação e regressão. As técnicas principais deste tipo de aprendizado são as Árvores de Decisão, *Support Vectors Machines (SVMs)* e as Redes Neurais (MAIMON; ROKACH, 2005; MONARD; BARANAUSKAS, 2003).

2.2 Redes Neurais Artificiais

Redes Neurais Artificiais (do inglês, *Artificial Neural Networks (ANN)*) são uma abordagem utilizada para o desenvolvimento de funções de aproximação de valores tanto reais quanto discretos, sendo muito utilizada em problemas de aprendizado, tais como, interpretação de dados de um sensor do mundo real.

O conceito de ANN foi criado utilizando como base o cérebro humano, com seus milhões de neurônios interconectados. Assim como o cérebro se comunica através das sinapses que ocorrem entre os neurônios, as ANNs buscam solucionar um problema pela comunicação entre seus **Perceptrons**, que são a essência de uma ANN e podem conter de 1 até vários perceptrons em uma única ANN. Perceptron pode ser definido como a menor estrutura de uma ANN, que recebe n valores de entrada (que podem ser a saída de outros perceptrons) e resultam em um único valor de saída (que pode ser a entrada de outro(s) perceptron(s)) (MITCHELL, 1997).

Um dos principais algoritmos de ANN utilizados é conhecido como *backpropagation*. Este algoritmo utiliza uma rede *multilayer feed-forward*, que consiste de uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Os atributos de uma amostra entram simultaneamente na camada de entrada e transmitidos para a primeira camada oculta, até que a última camada alimenta a camada de saída, que finalmente propaga a resposta final do modelo (HAN; KAMBER; PEI, 2011).

No algoritmo de *backpropagation*, o aprendizado ocorre iterativamente. Os atributos de uma amostra entram na rede, e após serem processados, o resultado é comparado à classe real da amostra. Para cada amostra então, os pesos das conexões entre os perceptrons são atualizados, de forma a reduzir o erro quadrático médio. Entretanto, esta atualização é feita de trás pra frente, começando da camada de saída até a primeira camada oculta. Não é garantido que os pesos irão convergir, porém em

geral isso acontece e o aprendizado termina. Mais detalhes podem ser encontrados em HAN; KAMBER; PEI (2011) juntamente com um pseudo-algoritmo.

2.2.1 Redes Neurais Recorrentes

Os algoritmos clássicos de *feedforward* são capazes de resolver problemas em diversos domínios diferentes. Entretanto, eles não são imunes à limitações. Uma das principais suposições das redes neurais clássicas é a de que cada amostra é individualmente independente das demais. Esta suposição pode, e é, verdade para diversas aplicações. Isto causa um grande problema quando as amostras a serem propagadas pela rede, tanto para treinamento quanto para teste, são de alguma relacionadas em tempo e/ou espaço. Este problema acontece porque nas técnicas clássicas, ao final da propagação de uma entrada, o estado inteiro da rede é perdido, sendo nenhuma informação sobre a entrada anterior armazenada e levada em consideração de alguma forma para as próximas. Ocorre somente a alteração dos pesos causada pelo erro na saída da rede. Isso quando há erro, senão a rede segue intacta.

Apesar das limitações de redes *feedforward*, elas possuem uma representatividade delas alta. Inclusive, a suposição de independência entre as amostras que concedeu tal poder ao longo dos anos. Diante disto, LIPTON; BERKOWITZ; ELKAN (2015) traz a indagação se de fato é necessário, de forma explícita, criar modelos sequenciais. Neste mesmo trabalho é respondida tal dúvida. Apesar da possibilidade de, implicitamente, mascarar os dados como sequenciais em uma rede clássica, como por exemplo, concatenando a entrada com os dados de seu predecessor e sucessor como é apresentado em MAAS et al. (2012). Esta abordagem limita a rede para uma janela de tempo finita, logo, não é possível levar em consideração dados anteriores à janela dada como entrada. Por fim, LIPTON; BERKOWITZ; ELKAN (2015) afirma que sem um modelo de sequenciamento explícito, é improvável que qualquer combinação de classificadores e regressores possa ser usada para simular tal característica.

Redes Neurais Recorrentes (do inglês, *Recurrent Neural Networks (RNN)* são a proposta para resolver o problema de continuidade temporal. RNNs são basicamente redes *feedfoward*, porém seus *perceptrons* são capazes de, com suas saídas, alimentar a si mesmo e aos outros perceptrons da mesma camada. Assim sendo, no momento t , nodos com entradas recorrentes recebem como entrada os dados da amostra $x^{(t)}$ e também as saídas dos nodos ocultos $h^{(t-1)}$. A saída $\widehat{y}^{(t)}$ para cada momento t é calculada a partir dos valores dos nodos ocultos h^t no momento t. A saída no momento $t-1$ pode influenciar a saída $\widehat{y}^{(t)}$ no momento t e nos momentos futuros através das conexões recorrentes.

Há duas equações principais para descrever o comportamento do exemplo apresentado na Figura 1:

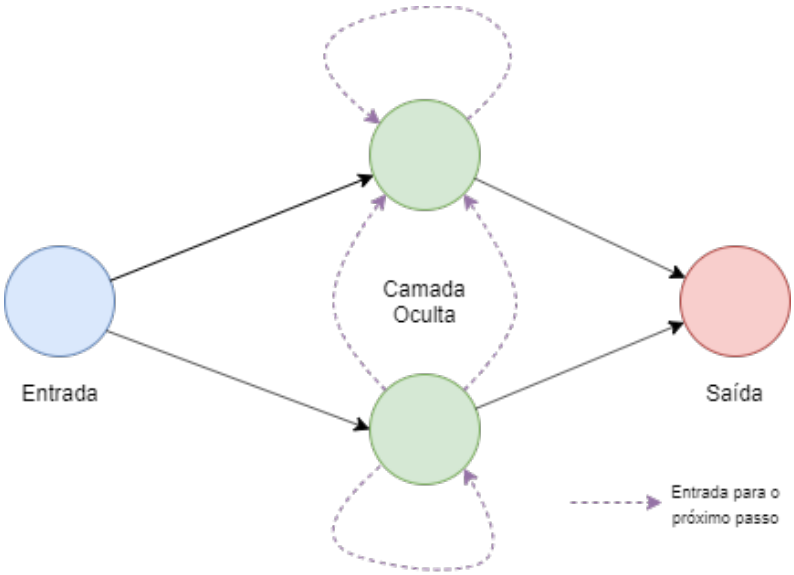

Figura 1 – Rede Neural Recorrente simples.

$$h^{(t)} = \sigma(W^{hx}x^{(t)} + W^{hh}h^{(t-1)} + b_h)$$

$$\widehat{y}^{(t)} = softmax(W^{yh}h^{(t)} + b_y)$$

Esta passagem de valores de um momento para outro permite que haja uma dependência sequencial nas amostras apresentada para a rede. Entretanto, esta característica torna o aprendizado da rede mais complexo, como mostra LIPTON; BERKOWITZ; ELKAN (2015). De forma simplificada, o algoritmo de *backpropagation* torna-se complexo, sendo necessário ajustes de pesos considerando não somente a propagação da entrada atual, pois o estado da rede no momento anterior influenciou o erro causado neste momento. Para resolver isto, foi proposta por WILLIAMS; ZIPSER (1989) uma solução chamada *Truncated Backpropagation Through Time (TBTT*. TBTT basicamente define um número finito de passos onde o erro pode ser propagado. Este corte no algoritmo pode aliviar o problema, porém sacrifica a capacidade da rede de aprender dependência de longo prazo.

Novos modelos de RNN foram propostos e dois se destacaram como mais bem sucedidos. São as redes LSTM (*Long-Short Term Memory*) e BRNN (*Bidirectional Recurrent Neural Networks*), propostas por HOCHREITER; SCHMIDHUBER (1996) e SCHUSTER; PALIWAL (1997) respectivamente. Uma rede BRNN, resumidamente, está estruturada de forma que seus *perceptrons* nas camadas ocultas são capazes de serem alimentados com dados dos momentos anteriores, como as RNNs padrões, assim como dos momentos posteriores. Este comportamento causa a bidirecionalidade

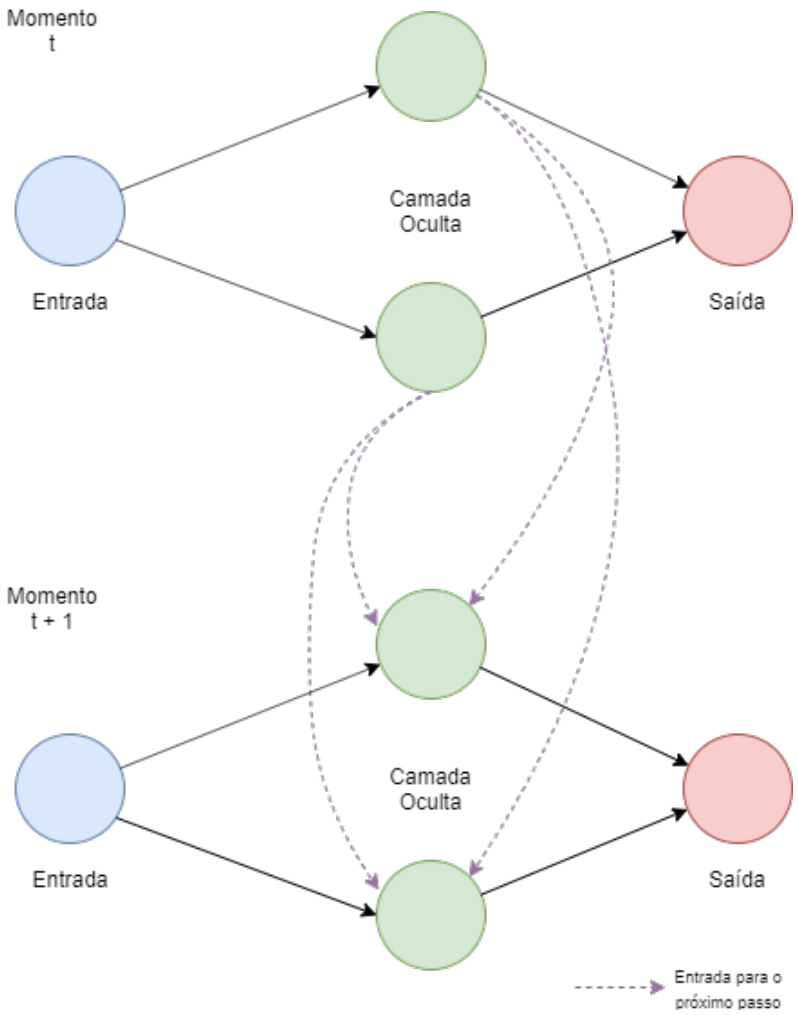

Figura 2 – Rede apresentada na Figura 1 em dois momentos: t e $t + 1$.

no aprendizado da rede.

Uma rede LSTM tem como objetivo remover o desaparecimento dos gradientes que ocorre nas RNNs comuns. Para realizar tal tarefa, cada *perceptron* comum da rede é substituído por uma célula de memória (do inglês, *memory cell*. Cada uma destas células contém um nodo recorrente com peso fixo em 1, garantindo que a informação passará por muitos momentos diferentes sem desaparecer. O presente trabalho faz uso de redes LSTM, por este motivo, o funcionamento da mesma será apresentado com maiores detalhes a seguir.

2.2.1.1 Long Short Term Memory

As redes *Long Short Term Memory* foram introduzidas primeiramente por HOCHREITER; SCHMIDHUBER (1996) com o objetivo de reduzir o desaparecimento dos gradientes passados de amostra em amostra durante o treinamento da rede, como explicado na seção anterior. *Long Term Memory* é uma característica herdada das redes RNN comuns, representada através de pesos, que vão atualizando vagaro-

samente, codificando o conhecimento adquirido para próximos momentos no tempo. *Short Term Memory* também é uma característica presente nas RNNs comuns, que ocorre na forma de ativações que passam de cada *perceptron* para os seus sucessores. As redes LSTM introduzem uma forma intermediária de guardar informação entre diferentes momentos, conhecido como *memory cell*.

Memory cells são os *perceptrons* de uma LSTM. São construídas a partir de nodos mais simples conectados em um padrão específico com a inclusão de nodos multiplicadores, representados por Π. Todos os elementos de uma *memory cell* estão descritos abaixo:

1. *Input node* - Este nodo, denominado g_c, é um nodo que recebe a ativação da camada de entrada $x^{(t)}$ no momento presente e da camada oculta no momento anterior $h^{(t-1)}$. Geralmente, a entrada passa por uma função de ativação *tanh*, embora no trabalho original tenha sido proposto o uso de *sigmoid*.

2. *Input gate* - *Gates* são uma característica implementada exclusivamente pelas redes LSTM. Similar ao *input node*, este *gate* recebe a ativação (*sigmoid*) da camada de entrada $x^{(t)}$ no momento presente, assim como da camada oculta no momento anterior. *Gates* tem este nome porque se seu valor for 0, o valor do outro nodo é desconsiderado. Já se seu valor for 1, o valor do outro nodo passa completamente inalterado. O *input gate* i_c é multiplicado pelo *input node*.

3. *Internal state* - No centro de toda *memory cell* há um nodo s_c com uma ativação linear. Este nodo é chamado de estado interno (do inglês, *internal state*) no trabalho original. Este nodo possui um vértice recorrente auto-conectado com peso fixo. Por causa do peso fixo, o erro percorre a rede sem desaparecer resolvendo o problema presente nas RNNs comuns. A atualização do *internal state* se dá através da equação:

$$s^{(t)} = g^{(t)} \odot i^{(t)} + s^{(t-1)}$$

   onde $\odot$ é a multiplicação ponto a ponto dos vetores.

4. *Forget gate* - Estes *gates* foram introduzidos por GERS (2001). Basicamente, eles tem como função permitir que a rede aprenda a fazer um *reset* da informação no *internal state*. Esta função é especialmente útil em redes que rodam continuamente, como explicado em LIPTON; BERKOWITZ; ELKAN (2015). Com estes *gates*, a equação do *internal state* se dá conforme:

$$s^{(t)} = g^{(t)} \odot i^{(t)} + f^{(t)} \odot s^{(t-1)}$$

5. *Output gate* - O *Output gate*, chamado aqui de v_c, é similar ao *input gate*, diferenciando-se apenas pelo fato de ser multiplicado pelo resultado do *internal state* ao invés de ser multiplicado pelo *input node*. O resultado desta multiplicação é a saída final da *memory cell*.

Note que a notação de vetor utilizada nas equações acima são referentes aos valores de todos as *memory cells* de uma camada inteira. Por exemplo, s refere-se a um vetor com os valores de todos s_c de cada *memory cell* c na camada. Quando é c está subscrito, refere-se somente a uma *memory cell* única.

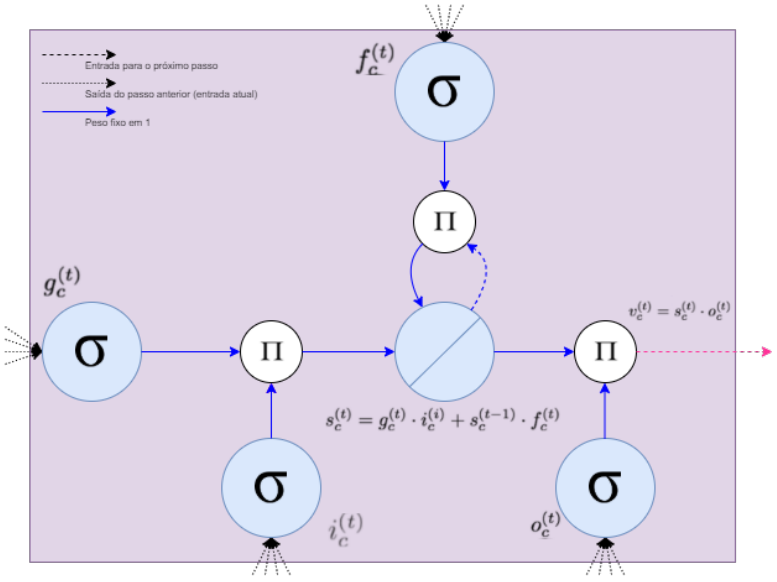

Figura 3 – *Memory cell* com *forget gate* apresentada em GERS (2001).

Por fim, a execução de um modelo LSTM segue as seguintes equações calculadas a cada momento. Tais equações compõem o algoritmo completo de uma LSTM moderna com *forget gates*:

$$g^{(t)} = \sigma(W^{gx}x^{(t)} + W^{gh}h^{(t-1)} + b_g)$$

$$i^{(t)} = \sigma(W^{ix}x^{(t)} + W^{ih}h^{(t-1)} + b_i)$$

$$f^{(t)} = \sigma(W^{fx}x^{(t)} + W^{fh}h^{(t-1)} + b_f)$$

$$o^{(t)} = \sigma(W^{ox}x^{(t)} + W^{oh}h^{(t-1)} + b_o)$$

$$s^{(t)} = g^{(t)} \odot i^{(t)} + s^{(t-1)} \odot f^{(t)}$$

$$h^{(t)} = \sigma(s^{(t)}) \odot o^{(t)}$$

Onde $h^{(t)}$ é o vetor de valores da camada oculta no momento t, e $h^{(t-1)}$ corresponde aos valores de saída de cada *memory cell* na camada oculta no momento anterior.

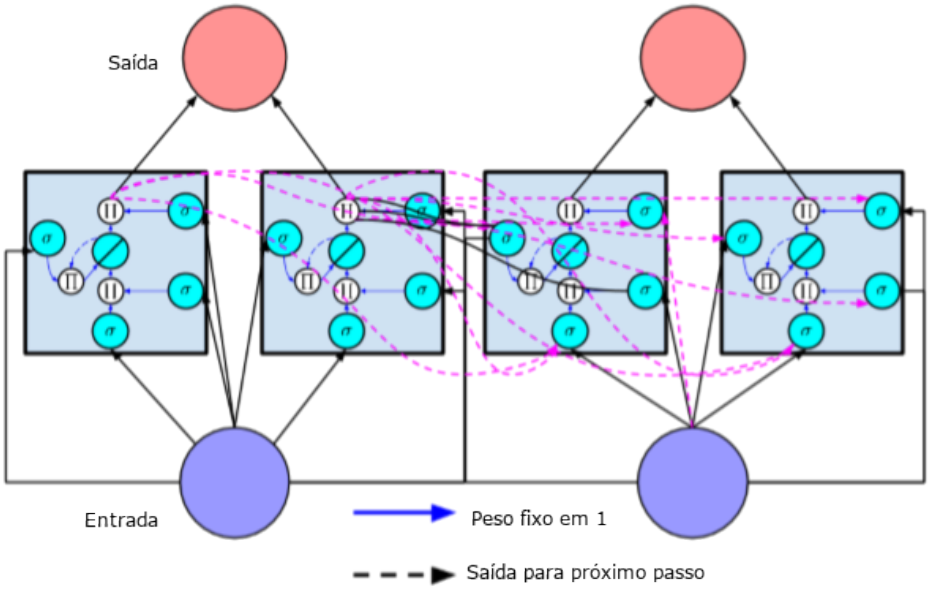

Figura 4 – Rede LSTM com uma camada oculta contendo duas *memory cells*. Execução da rede em dois momentos sequenciais (LIPTON; BERKOWITZ; ELKAN, 2015).

3 TRABALHOS RELACIONADOS

Neste capítulo serão apresentados trabalhos relacionados ao tema deste trabalho. Entre os trabalhos descritos a seguir, encontram-se trabalhos que buscaram melhores resultados para suas aplicações a partir do uso de um modelo de aprendizado online e também trabalhos que buscaram identificar e então descrever o crescimento da população de diferentes insetos e ocorrência de doenças nas plantações.

Como esta é uma área de atuação da computação aplicada e que diz respeito a um assunto de importância para a economia de muitos países, é comum encontrar trabalhos que buscam o mesmo objetivo através do uso de técnicas à parte das utilizadas pela computação. Por este motivo, a grande maioria dos trabalhos encontrados fazem uso de modelos estatísticos para descrever o crescimento das pragas. Modelos probabilísticos, regressões lineares e polinomiais estão entre as mais frequentes técnicas utilizadas por autores de áreas diferentes da computação. Entretanto, é possível encontrar alguns poucos trabalhos que utilizam, principalmente, redes neurais como técnica de previsão de infestações. A seguir pode-se encontrar detalhes sobre os principais trabalhos encontrados.

Em HU et al. (2018) uma rede LSTM é implementada como uma abordagem para melhorar o Reconhecimento de Voz Automático (do inglês, *Automatic Speech Recognition*). Os autores do trabalho propõe a criação de uma rede com aprendizado online incremental de forma individual, ou seja, a rede é capaz de aprender dinamicamente e melhorar seu desempenho quando usada por um único indivíduo somente. Durante os experimentos, os autores utilizaram um modelo previamente treinado de forma offline como base para o modelo online. Para o treinamento da rede, a cada segmento contendo x entradas, a rede era treinada com este novo mini *batch* gerando assim um novo modelo que era avaliado no próximo segmento. Ao final dos experimentos, os autores concluem que o modelo proposto obteve uma redução média de 19,18% na perplexidade do modelo e uma redução de 2,8% na taxa de erro de palavras.

Em OZAY et al. (2015) são apresentadas diferentes técnicas de Aprendizado de Máquina para detecção de ataques em *Smart Grids*. Entre as técnicas apresentadas, é proposto o uso de aprendizado online (Perceptron, Perceptron com pesos, SVM

Online e SLR Online), aprendizado supervisionado (SVE, Perceptron, Linear SVM, KNN e SLR), aprendizado semi-supervisionado (S3VM) e algoritmos de fusão em nível de decisão e recurso (AdaBoost e MKL). Para o aprendizado online, a cada nova amostra, os modelos são retreinados com a nova amostra e seus pesos são então atualizados. Destaca-se que os algoritmos utilizados para o aprendizado online são os mesmos utilizados no aprendizado supervisionado. Ao final dos experimentos, os autores concluem afirmando que os resultados dos modelos online são equivalentes aos resultados dos modelos supervisionados, porém com redução na complexidade computacional dos algoritmos.

MOULY; SHIVANANDA; VERGHESE (2017) propõe um trabalho onde é desenvolvido um modelo para prever a ocorrência de *Bactrocera dorsalis*, uma espécie de mosca-da-fruta, em um pomar de mangas. Para a obtenção dos dados populacionais das moscas, foram utilizadas armadilhas estrategicamente disponibilizadas pelo pomar para a captura de amostras. A contagem de moscas capturadas foi feita semanalmente. Dados climáticos foram coletados diariamente de uma estação meteorológica próxima ao pomar. Os dados coletados correspondem à temperatura máxima e mínima, umidade relativa do ar às 7:30 e as 13:30, nível de chuva e velocidade do vento. A partir dos dados coletados, foram realizados testes independentes para cada variável climática, em busca das que mais se correlacionavam com o número de moscas coletadas. A partir dos resultados foi possível perceber que temperatura máxima e mínima, velocidade do vento e nível de chuva foram as variáveis que obtiveram melhores resultados de correlação. O autor criou um modelo linear de previsão de uma única variável para cada variável climática para verificar se o número de moscas dado pelo modelo diferia do número presente nas armadilhas. Como resultado, foi constatado que para cada um dos quatro modelos a previsão realizada não tinha uma diferença significante do número real de moscas, através do teste t de student. Por fim, o autor desenvolve um modelo regressivo que leva em conta todas as quatro variáveis mais correlacionadas para fazer a previsão de infestações, além de obter informações valiosas com relação à variação da população de acordo com cada variável.

Em VENNILA et al. (2017) é apresentado um trabalho cujo objetivo é investigar o potencial do uso de técnicas de computação leves como Redes Neurais Polinomiais e Artificiais para prever os picos de ocorrência de *Spodoptera litura* em plantações de amendoim utilizando variáveis climáticas, comparando aos resultados obtidos com o uso de modelos de regressão estatística que é a técnica tradicionalmente utilizada. Para a realização deste trabalho, foram utilizados dados populacionais dos insetos, que foram coletados semanalmente através de armadilhas colocadas nas plantações no decorrer de 25 anos (1990-2014). Além disso, foram utilizados dados climáticos obtidos no observatório meteorológico presente no mesmo local das plantações. Os dados climáticos utilizados foram temperatura máxima e mínima, nível de chuva e

umidade relativa durante o período da manhã e da tarde. Foi decidido pelos autores prever qual o momento de pico de ocorrência da praga, para isso, foi encontrada a semana do ano em que o pico ocorre e quatro modelos diferentes foram treinados com os dados meteorológicos das duas semanas anteriores ao pico. Os quatro modelos são: Regressão Linear Múltipla, Regressão Polinomial, Rede Neural Artificial (MLP) e Rede Neural Polinomial. Após os testes realizados nas 25 estações disponíveis na base de dados, foi constatado que a Regressão Linear Múltipla conseguia explicar 52% dos casos, enquanto que Regressão Polinomial alcançou a marca de 81%, Rede Neural Polinomial, 73%, e Rede Neural Artificial (MLP), 88%. A partir desdes resultados, pode-se concluir que os algoritmos de redes neurais são melhores para entender a relação existente entre a população de *S. litura* e as condições climáticas. Já para fazer a previsão da população de insetos, a Rede Neural Artificial (MLP) destaca-se em relação às outras técnicas utilizadas.

No trabalho desenvolvido por CALAMA et al. (2017), é apresentada uma modelagem da dinâmica da infestação de *Dioryctria mendacella* em pinhas da espécie *Pinus pinea*. Para a modelagem, foram criadas 52 áreas de coleta. Cada área corresponde à uma região circular que contém 10 árvores. Durante os anos de 1996 e 2005, as pinhas dessas árvores foram coletadas e classificadas como saudáveis ou danificadas. Por sua vez, as danificadas foram divididas em 3 categorias: 1) as pinhas atacadas pela espécie *D. mendacella* , 2) as pinhas atacadas pela espécie *P. validirostris* e 3) as pinhas atacadas por ambas espécies. Como este trabalho tinha interesse somente na espécie *D. mendacella*, as pinhas da categoria 2 foram descartadas. Foi decidido então a realização de análises para cada área em cada determinado ano, entre 1996 e 2005. Para realizar a previsão de infestações, foi realizada uma análise exploratória nos dados. Nesta análise, foi constatado que existia uma enorme variabilidade de pinhas afetadas pela praga entre os anos de coleta, assim como uma alta variabilidade de cones coletados em cada ano. A média da proporção de pinhas afetadas foi de 17,1%, variando entre 7,1% (2008) e 32,4% (2001). A partir da análise exploratória foi constatado também que o uso dos dados do número pinhas afetados e/ou número total de pinhas coletadas no ano anterior são relevantes para a criação do modelo de previsão de pinhas afetadas no ano atual. Ao fim dos testes realizados, foi percebido que para a estimação da probabilidade de dano em uma única pinha em uma dada área e ano é influenciada pela altitude e pela área basal da região. Considerando os dados climáticos, a média da temperatura máxima diária de dezembro à fevereiro e de maio à junho são as variáveis que mais se destacaram no modelo. Além disso, o número de pinhas afetadas no ano anterior tem uma alta influência no número de pinhas afetadas no ano corrente. Por fim, quanto maior o número de pinhas produzidas em uma área em um ano, menor a chance de uma pinha ser afetada. Para a previsão de pinhas afetadas, foram criados dois complexos modelos probabilísticos que o

autor chama de *conditional* e *marginal*. A eficiência dos modelos para o número de pinhas afetadas por área foi de 0,74 (*marginal*) e 0,95 (*conditional*), enquanto que para a probabilidade de uma pinha ser afetada foi de 0,45 (*marginal*) e 0,66 (*conditional*).

4 APRENDIZADO ONLINE APLICADO NA PREVISÃO DE INFESTAÇÃO DE INSETOS UTILIZANDO UMA REDE *LSTM*

4.1 Base de Dados

A base de dados utilizada para este trabalho foi a mesma base utilizada em RAGA et al. (2017). Conforme descrito pelos autores, esta base é formada por dados coletados entre os anos de 2004 e 2006. O controle populacional foi realizado sobre duas espécies principais: *Ceratitis capitata* e *Anastrepha fraterculus*. Os dados foram coletados de árvores do Banco de Germoplasma de Árvores Frutíferas do Instituto Agronômico de Campinas. Tais árvores ficam localizadas em Capão Bonito, sudoeste do estado de São Paulo. Esta região é caracterizada por possuir clima subtropical úmido. Durante o experimento realizado, não foram aplicados nenhum tipo de pesticida na região. Os dados climáticos da base foram coletados de uma estação meteorológica localizada à aproximadamente 250 metros da área das coletas do insetos.

A base de dados meteorológicos é composta das seguintes variáveis: Temperatura (°C), Nível de Chuva, Armazenamento, Evapotranspiração, Deficit Hídrico e Excedente Hídrico. O dado de Armazenamento não foi esclarecido o significado e não consta em RAGA et al. (2017). Além disto, cada dado corresponde à média de cada variável durante o período de uma semana, sendo a primeira semana de 12/01/2004 à 18/01/2004.

Os dados relacionados à população de insetos foram coletados utilizando 10 armadilhas Mc Phail com atrativo alimentício. Este atrativo era renovado semanalmente, quando foi feita a contagem dos insetos coletados para a semana e o piso da arma-

Tabela 1 – Exemplos dos primeiros 5 (cinco) elementos da base de dados meteorológicos.

Temperatura	Chuva	Armazenamento	Evapotranspiração	Deficit Hídrico	Excedente Hídrico
22,3	5	105	25	2	0
21,3	70	125	25	0	40
22	223	125	25	0	198
23,2	54	125	26	0	28
22,3	61	125	25	0	36

dilha era substituído por um novo. Cada semana de coleta corresponde ao final da semana de coleta de dados meteorológico. De acordo com RAGA et al. (2017), a captura de fêmeas é superior à captura de machos. Entretanto, para a execução deste trabalho, não é feita a distinção entre o gênero dos insetos.

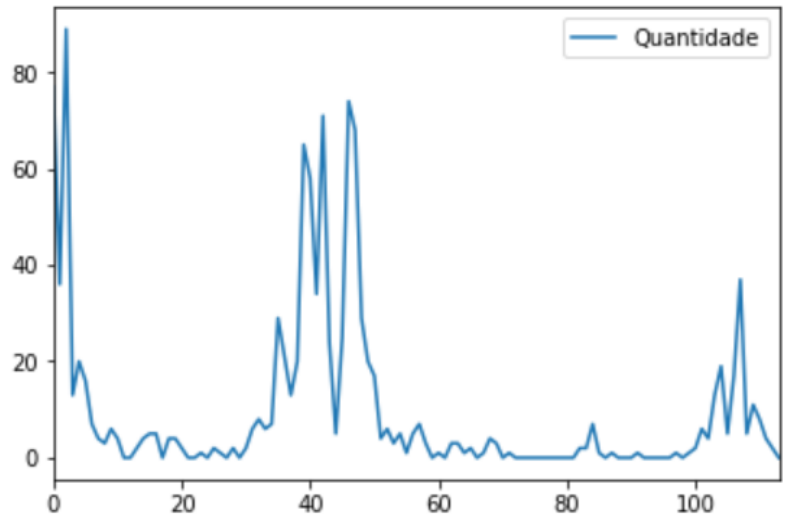

Figura 5 – Quantidade de *Anastrepha fraterculus* coletadas semanalmente entre 2004 e 2006.

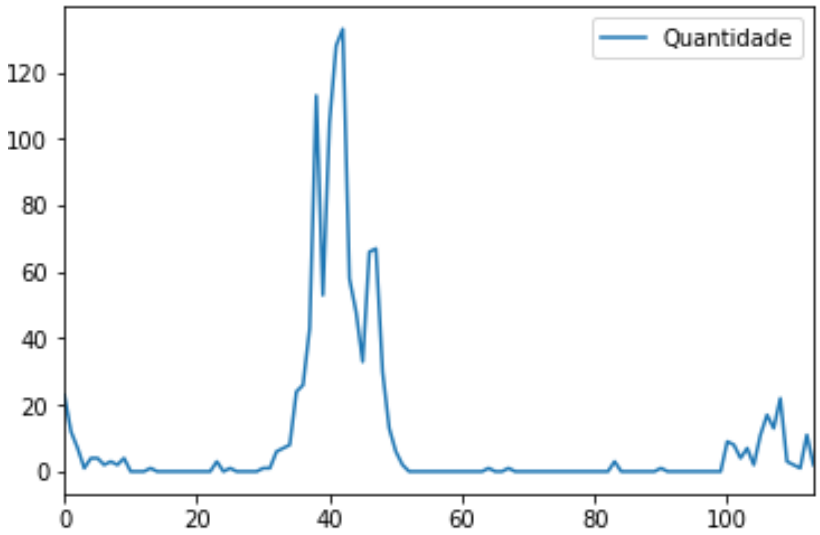

Figura 6 – Quantidade de *Ceratitis capitata* coletadas semanalmente entre 2004 e 2006.

4.2 Pré-Processamento dos Dados

O pré-processamento pode ser considerado uma das mais importantes etapas no processo de descoberta de conhecimento. Em casos reais, é extremamente comum

encontrar bancos de dados com informações faltando ou incorretas, ou até mesmo encontrar os dados espalhados em diversos *datasets* (HAN; KAMBER; PEI, 2011).

Durante esta etapa, não foram encontrados casos graves que necessitassem de trabalho extensivo para a resolução do problema. Haviam pouquíssimos casos onde os dados meteorológicos estavam faltando para uma determinada semana, o que, para este projeto, é bastante significativo pois inclui uma lacuna no treinamento do modelo. Para resolver este problema, foi calculada a média aritmética das variáveis meteorológicas da semana anterior e próxima.

Não foram encontrados casos de *outliers* na base. Quando há a ocorrência de *outliers*, eles geralmente representam um erro de gravação na base ou algum valor de leitura indevido nos sensores.

Após o processo descrito acima (conhecido como Limpeza dos Dados), foi realiza a Seleção de Dados, onde as variáveis são analisadas e então, conforme a avaliação, são removidas ou criadas novas a partir das já existentes. Nesta etapa, foi-se decidido remover a variável de Armazenamento disponível na base. O principal fator para esta decisão foi o desconhecimento da informação representada por este dado e, portanto, a não capacidade de replicação para futuras bases de dados.

A última etapa de pré-processamento realizada foi a Normalização e a Uniformização dos Dados.

4.2.0.1 Uniformização

A uniformização dos dados é uma tarefa comum de ser aplicada antes do treinamento do modelo. A técnica consiste em transladar os dados para a origem 0, removendo de cada valor a média calculada para aquele atributo, e escalonar os dados, dividindo cada valor dos atributos que não são constantes pelo seu desvio padrão. Isso acontece porque muitos algoritmos não reagem bem com características que não possuem distribuição aproximada de uma curva normal. Diversas funções objetivas de algoritmos de aprendizado assumem que os dados estejam na origem e tenham o mesmo grau de variância. Caso algum atributo possua uma variância muito maior que os outros, ele pode dominar o função, impedindo que o algoritmo aprenda com os outros atributos também.

4.2.0.2 Normalização

A normalização de dados é um processo que visa re-escalonar as amostras de forma independente, com o objetivo de totalizar 1 em sua função Norma (que pode ser L_1 ou L_2). A função Norma associa a cada vetor de um espaço vetorial um valor real não-negativo. Este valor é considerado o comprimento do vetor.

Seja $\vec{x} = (x_1, x_2, x_3, ..., x_n)$, então as funções Norma para este vetor, chamadas de **norma** L_p, é definida por:

$$\|\vec{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}, 1 <= p < \infty \quad (1)$$

$$\|\vec{x}\|_\infty = \max_{i=1}^{n} |x_i|^p \quad (2)$$

Existem 2 valores para p que destacam-se dos demais. Quando $p = 1$, a função Norma corresponde ao comprimento Manhattan da amostra, e quando $p = 2$, é calculado o comprimento Euclidiano.

4.2.1 Base de Treinamento

Feita a análise da base de dados obtida, constatou-se que ela possuía um total de 114 amostras. Após alguns testes iniciais, constatou-se que este número de amostras não era suficiente para o treinamento do modelo de aprendizado online, proposto neste trabalho. Portanto, foi decidido criar uma base de treinamento para o modelo a partir dos dados obtidos nesta base menor.

Para a criação da base de treinamento, optou-se por treinar uma rede LSTM com os dados da base descrita na seção 4.1, chamada a partir deste momento de base A. Com o modelo treinado, foi possível utilizar os dados climáticos da base utilizada em (SOUZA et al., 2017), chamada a partir deste momento de base B, e assim definir o número de insetos presente a cada semana. Esta nova base criada possui no total 1454 amostras.

Durante o tratamento dos dados, foi percebido que havia diferenças entre as duas bases meteorológicas. A base B não possuía as mesmas variáveis climáticas que a base A. Logo, foi necessário utilizar para o treinamento do modelo, somente as variáveis que eram comuns à ambas bases. São estas varáveis: temperatura e nível de chuva. Além destas duas variáveis, foram adicionadas três variáveis informando quantos insetos houve na semana da coleta dos dados meteorológicos (o que corresponde à saída da amostra diretamente anterior), e outras duas que foram criadas para gerar mais informações à rede, que são a temperatura multiplicada pelo nível de chuva e a diferença de insetos entre a semana atual e a anterior.

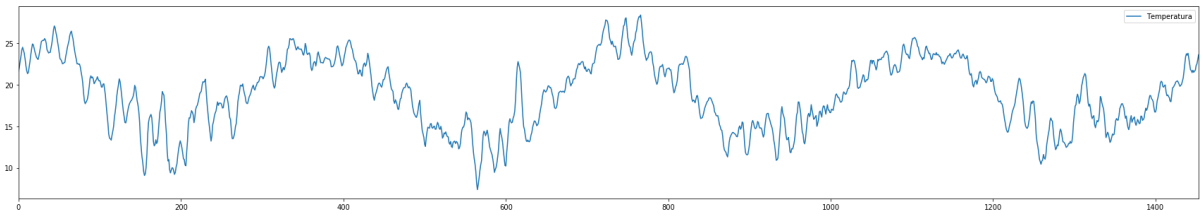

Figura 7 – Temperatura média das semanas presentes na base de dados B.

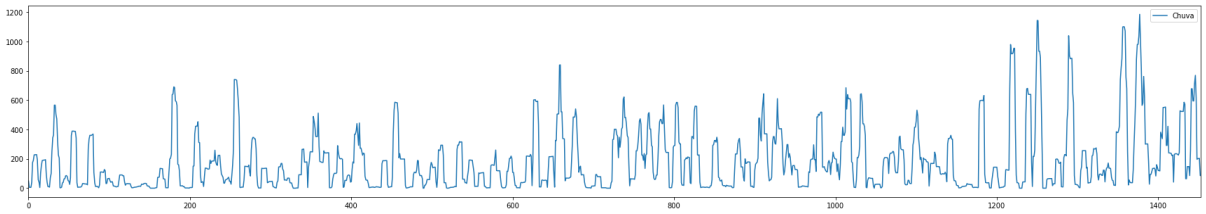

Figura 8 – Nível de chuva médio das semanas presentes na base de dados B.

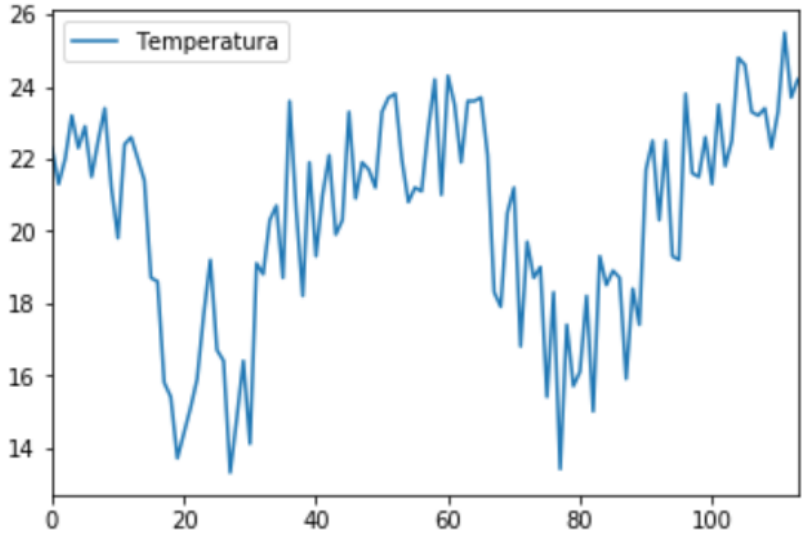

Figura 9 – Temperatura média das semanas presente na base de dados A.

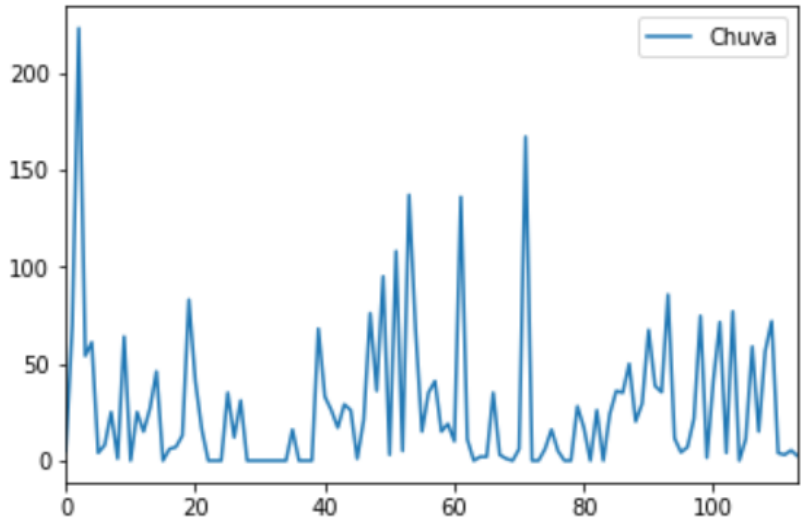

Figura 10 – Nível de chuva médio das semanas presentes na base de dados A.

A rede utilizada durante o treinamento é composta de um camada LSTM com 16 *memory cells*, com uma taxa de *dropout* de 0.3 e uma taxa de *dropout* recorrente de 0.3 para evitar *overfitting* do modelo. A camada de saída é composta por um

perceptron totalmente conectado às *memory cells* da camada anterior. Para o cálculo de *loss* foi utilizada a métrica Erro Médio Quadrático (do inglês, *Mean Squared Error (MSE)*) e o *optimizer* escolhido foi "adam".

Para o treinamento do modelo, foi utilizado 100% dos dados para treinamento, devido à baixa quantidade de amostras. Ao final do treinamento, o modelo obteve um erro médio quadrático de 236.63 para *Ceratitis capitata* e 146.89 para *Anastrepha fraterculus*, devido à picos encontrados na amostragem dos dados cuja rede não foi capaz de representar de forma adequada além de um visível atraso na rede para aprender por causa da baixa quantidade de dados.

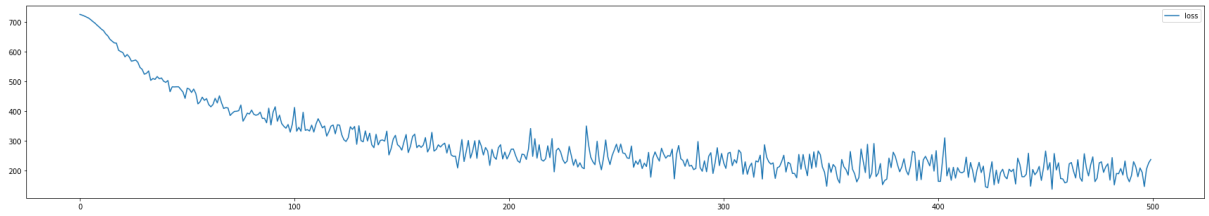

Figura 11 – Erro Médio Quadrático durante treinamento da rede LSTM para geração da base populacional de *Ceratitis capitata*.

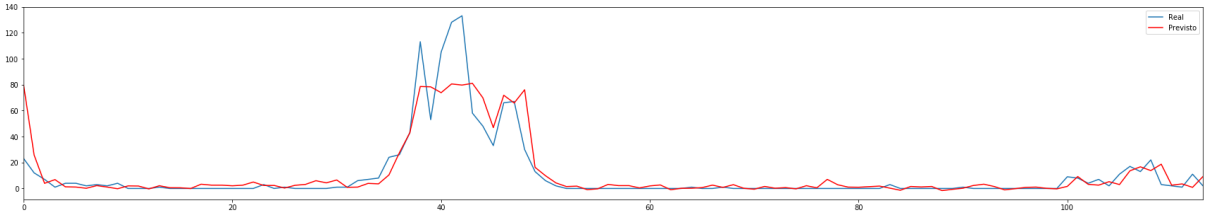

Figura 12 – Resultados obtidos após treinamento da rede LSTM sob o conjunto de treinamento de *Ceratitis capitata*.

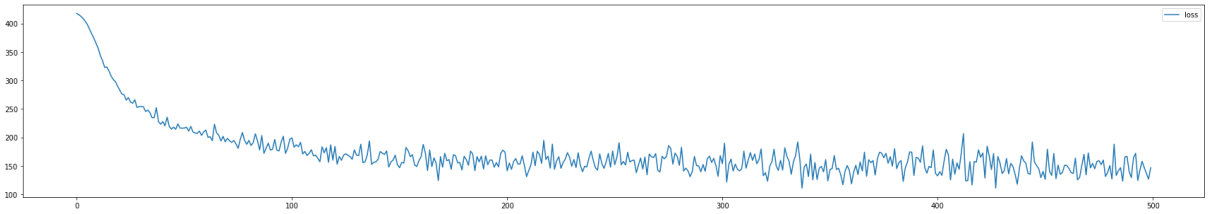

Figura 13 – Erro Médio Quadrático durante treinamento da rede LSTM para geração da base populacional de *Anastrepha fraterculus*.

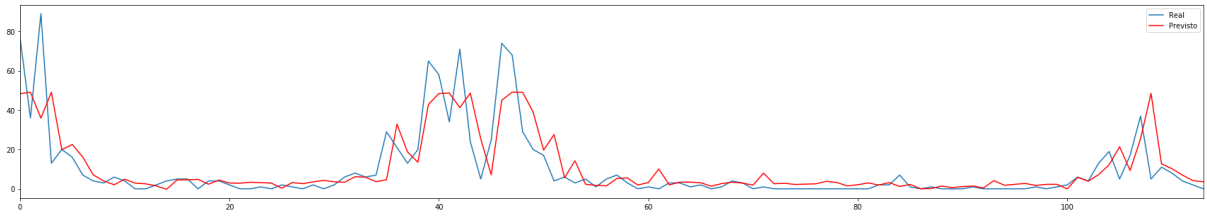

Figura 14 – Resultados obtidos após treinamento da rede LSTM sob o conjunto de treinamento de *Anastrepha fraterculus*.

Por fim, os dados da base B foram propagados pela rede LSTM criada e foram gerados assim os valores necessários para os conjuntos de treinamento para o modelo de aprendizado online conforme as Figuras 15 e 16 para *Ceratitis capitata* e *Anastrepha fraterculus*, respectivamente. Apesar de não refletirem dados reais da população destes insetos, acredita-se que desta maneira foi possível simular um ambiente próximo do real e demonstrar como um modelo de aprendizado online se comportaria para aprender a partir dos dados meteorológicos mesmo utilizando um *toy dataset*.

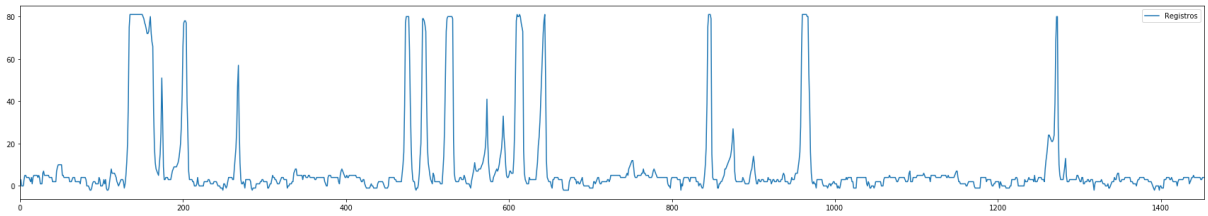

Figura 15 – Quantidade de *Ceratitis capitata* a cada semana prevista pela rede LSTM para o conjunto de treinamento.

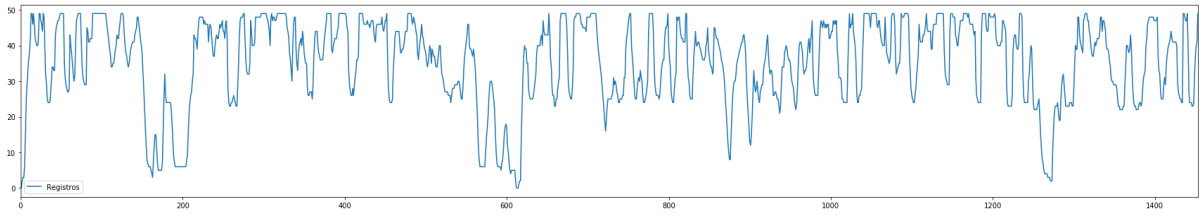

Figura 16 – Quantidade de *Anastrepha frateruculus* a cada semana prevista pela rede LSTM para o conjunto de treinamento.

4.2.2 Treinamento do Modelo de Aprendizado Online

A proposta deste trabalho é a construção de um modelo capaz de aprender dinamicamente com os dados, ou seja, um modelo de aprendizado online. Devido à natureza

do domínio desta aplicação, foi decidido utilizar redes neurais recorrentes devido às vantagens que elas apresentam conforme demonstrado no Capítulo 2.

Para a construção do modelo, foi utilizada a biblioteca Keras (CHOLLET et al., 2015). Keras é um biblioteca de redes neurais de alto nível desenvolvida em Python que é capaz de rodar sobre TensorFlow.

A rede neural escolhida para o treinamento online foi, assim como na geração do conjunto de treinamento, uma rede LSTM. A estrutura da rede utilizada neste modelo é, da mesma forma, bastante similar à utilizada na etapa anterior. Ela consiste de uma camada LSTM com 128 *memory cells*, um *dropout* de 0.3 e um *dropout* recorrente também de 0.3 para evitar *overfitting* do modelo. Por fim, a camada de saída possui apenas um perceptron totalmente conectado à camada anterior. A função de *loss* utilizada foi o MSE e o *optimizer* escolhido foi "adam".

Para o treinamento online do modelo, foram realizados múltiplas chamadas ao método *train_on_batch* do modelo, que é o método responsável por executar uma atualização única de gradientes em um único lote de dados. Para cada entrada da base de dados, foram executadas um total de 25 chamadas ao método *train_on_batch* para esta mesma entrada. Ao final do incremento do treinamento do modelo com esta entrada, foi feita a previsão de insetos para a próxima entrada HU et al. (2018).

Este procedimento buscou simular o comportamento de como seria o treinamento de um modelo em uma aplicação real. A cada semana seriam obtidos novos dados sobre a contagem de insetos presentes na plantação, desta forma seria possível o treinamento do modelo com os dados meteorológicos e populacional dos insetos da semana anterior. A Figura 17 mostra o fluxo descrito para treinamento e avaliação do modelo de aprendizado online em 2 passos.

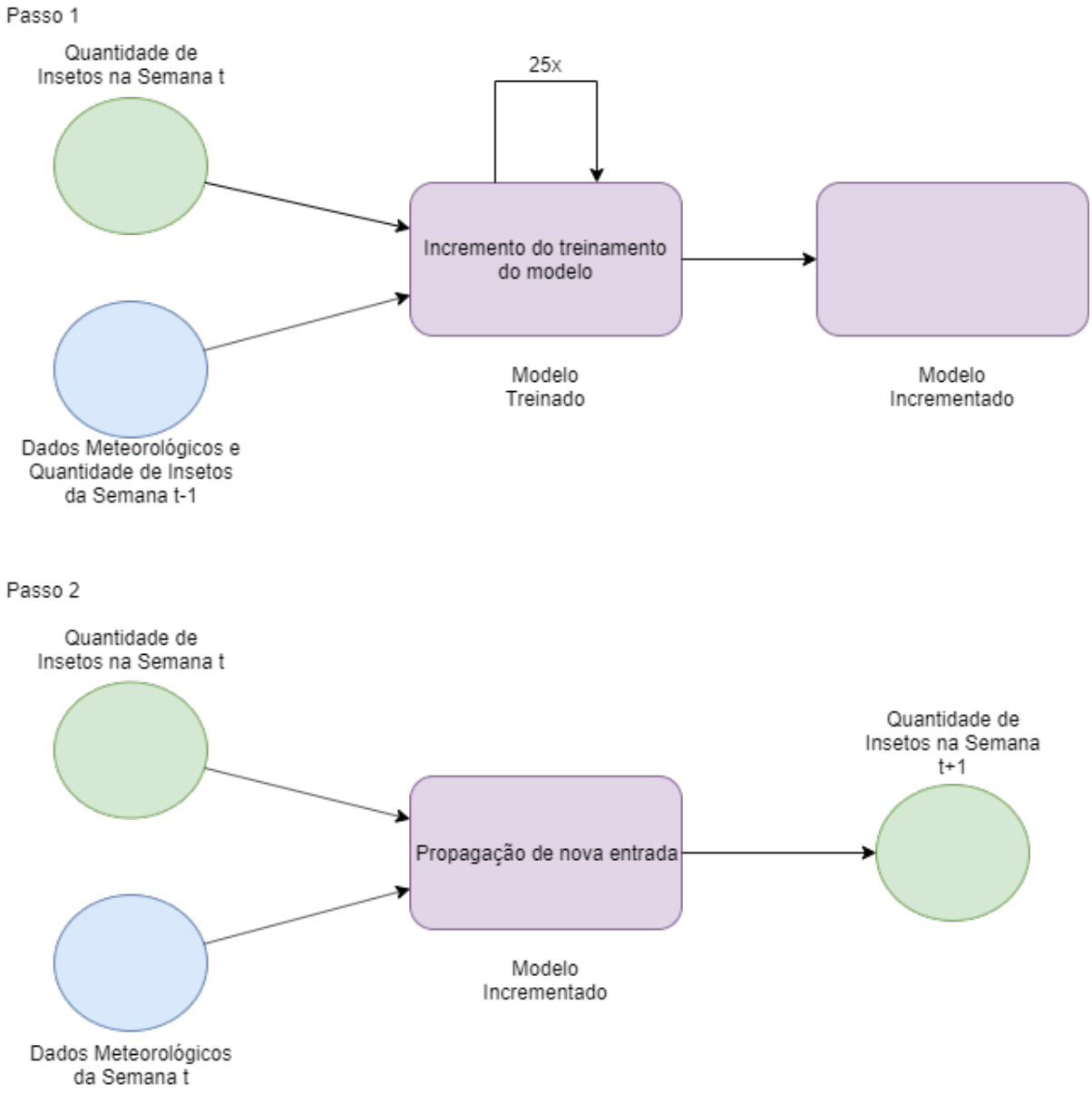

Figura 17 – Fluxo de execução do treinamento e avaliação do modelo de aprendizado online em dois passos.

Para comparação do modelo de aprendizado online, foi decidido utilizar a mesma base de dados para treinar uma rede LSTM da maneira tradicional, dividindo-a em dois conjuntos: treinamento e teste.

Para criação deste modelo, foi utilizada uma rede LSTM com a mesma estrutura do modelo anterior e com os mesmos hiperparâmetros. A base de dados foi dividida em 66% para conjunto de treinamento e 33% para conjunto de testes, respeitando a sequência temporal das amostras. O treinamento ocorreu durante 50 épocas.

4.2.3 Resultados e Discussão

Para avaliação dos resultados do modelo de aprendizado online foi utilizada a métrica do Erro Médio Quadrático (MSE). O MSE pode ser definido como uma métrica de fidelidade de sinal cujo objetivo é comparar dois sinais fornecendo uma pontuação quantitativa que descreve o grau de similaridade ou o grau de erro/distorção entre eles (WANG; BOVIK, 2009). Nesta métrica, geralmente um dos sinais é visto como original e o outro como distorcido ou contaminado por erros. O cálculo para o MSE se dá conforme a equação:

$$MSE(x, y) = \frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2. \tag{3}$$

Para o modelo de aprendizado online, o MSE foi calculado considerando todos os dados contidos na base, uma vez que o treinamento é gradual. Já para os modelos de aprendizado tradicional, o MSE foi calculado para os conjuntos de treinamentos, que foram propagados pela rede após a conclusão dos treinamentos, e para os conjuntos de testes, propagando novas entradas nunca antes vistas pela rede. Ao final dos testes, obteve-se os seguintes resultados:

Tabela 2 – Cálculo do Erro Médio Quadrático (MSE) para os modelos treinados.

	Aprendizado Online	**Aprendizado Tradicional (Treinamento)**	**Aprendizado Tradicional (Teste)**
Ceratitis capitata	37.44	53.71	4.70
Anastrepha fraterculus	23.07	29.89	13.66

Para as redes treinadas de forma tradicional, é possível analisar o comportamento do erro médio quadrático ao decorrer das 50 épocas em que foram treinadas nas Figuras 18 e 19.

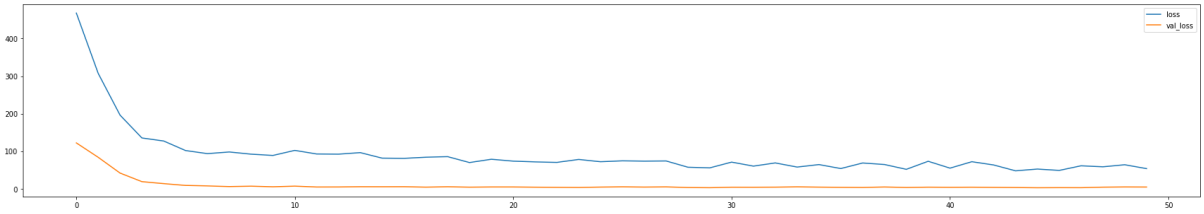

Figura 18 – Erro Médio Quadrático ao decorrer das 50 épocas de treinamento da rede para *Ceratitis capitata*.

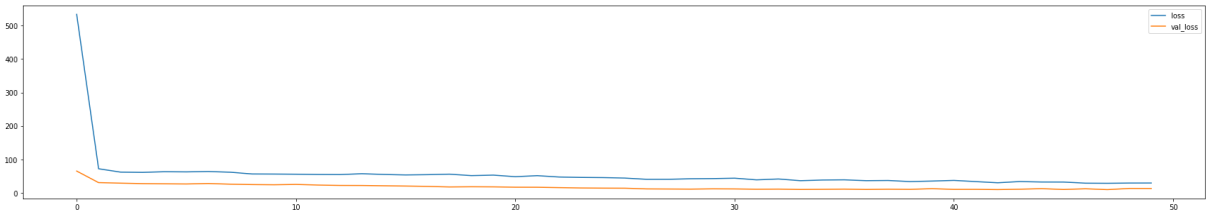

Figura 19 – Erro Médio Quadrático ao decorrer das 50 épocas de treinamento da rede para *Anastrepha fraterculus*.

Por fim, pode-se verificar o gráfico dos resultados contrapondo-se aos valores esperados como resposta para cada uma das espécies de interesse deste trabalho:

4.2.3.1 Ceratitis capitata

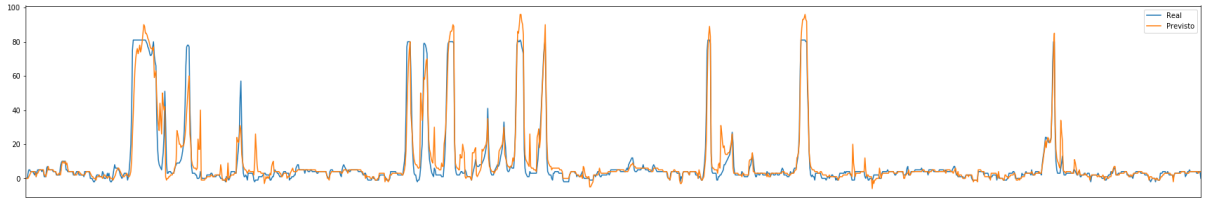

Figura 20 – Resultado da rede LSTM com aprendizado online comparado aos valores reais esperados.

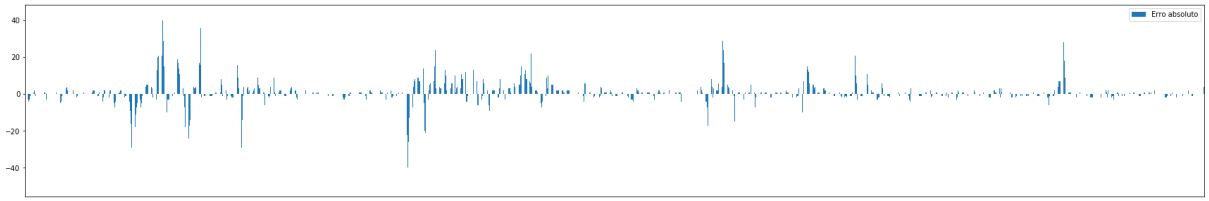

Figura 21 – Erro absoluto da rede LSTM com aprendizado online para cada entrada.

Figura 22 – Média do erro absoluto da rede LSTM com aprendizado online a cada 100 amostras.

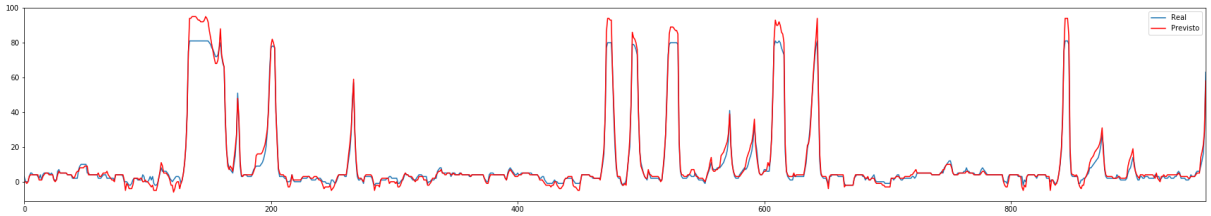

Figura 23 – Resultado da rede LSTM com aprendizado tradicional comparado aos valores reais esperados utilizando o conjunto de treinamento.

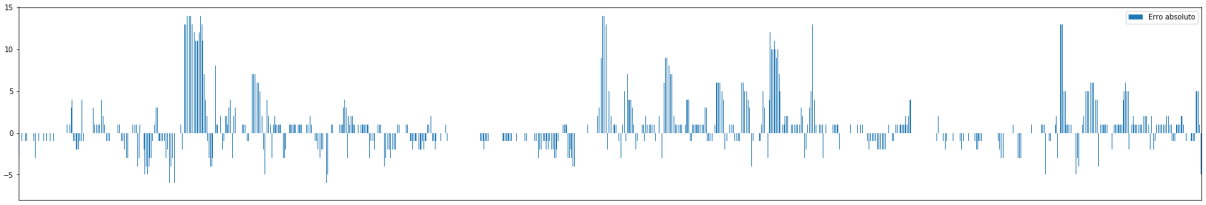

Figura 24 – Erro absoluto da rede LSTM com aprendizado tradicional para cada entrada utilizando o conjunto de treinamento.

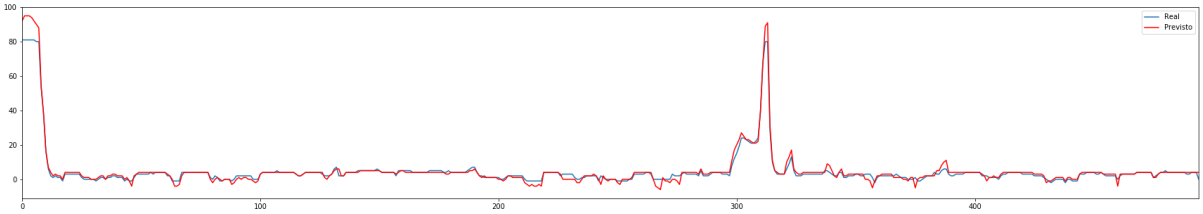

Figura 25 – Resultado da rede LSTM com aprendizado tradicional comparado aos valores reais esperados utilizando o conjunto de teste.

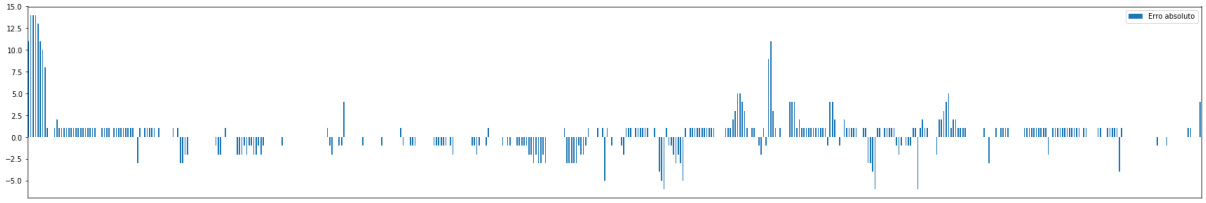

Figura 26 – Erro absoluto da rede LSTM com aprendizado tradicional para cada entrada utilizando o conjunto de teste.

4.2.3.2 Anastrepha fraterculus

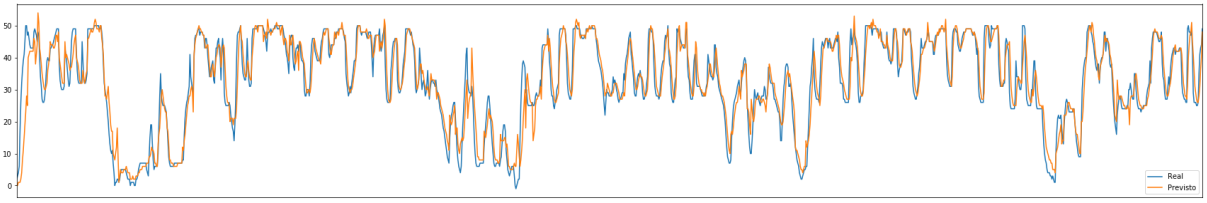

Figura 27 – Resultado da rede LSTM com aprendizado online comparado aos valores reais esperados.

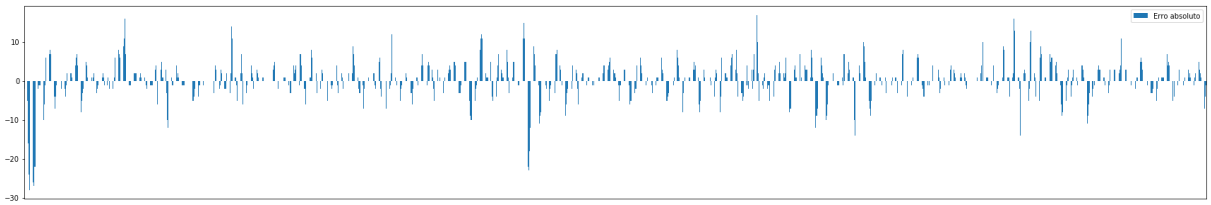

Figura 28 – Erro absoluto da rede LSTM com aprendizado online para cada entrada.

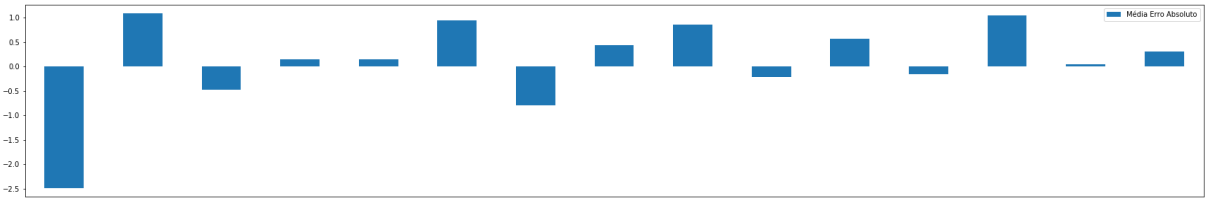

Figura 29 – Média do erro absoluto da rede LSTM com aprendizado online a cada 100 amostras.

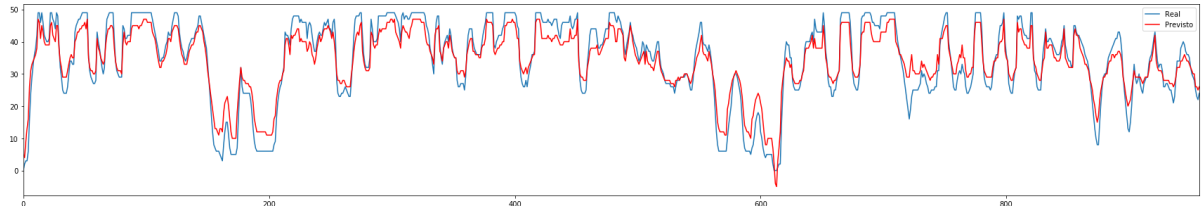

Figura 30 – Resultado da rede LSTM com aprendizado tradicional comparado aos valores reais esperados utilizando o conjunto de treinamento.

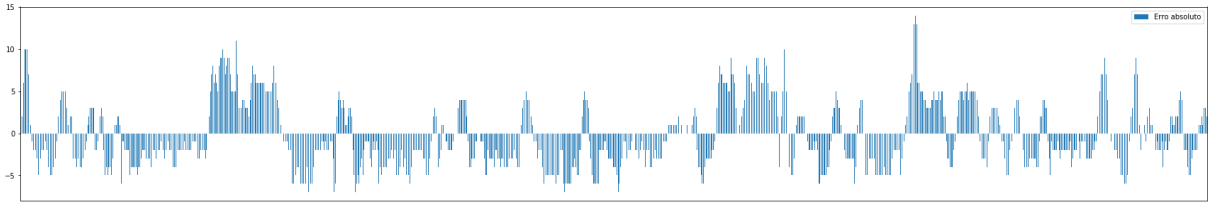

Figura 31 – Erro absoluto da rede LSTM com aprendizado tradicional para cada entrada utilizando o conjunto de treinamento.

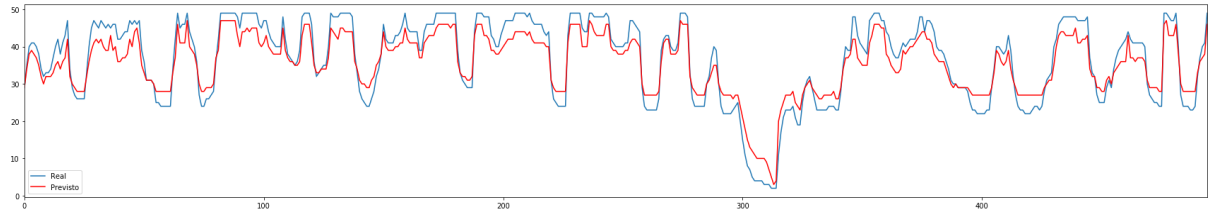

Figura 32 – Resultado da rede LSTM com aprendizado tradicional comparado aos valores reais esperados utilizando o conjunto de teste.

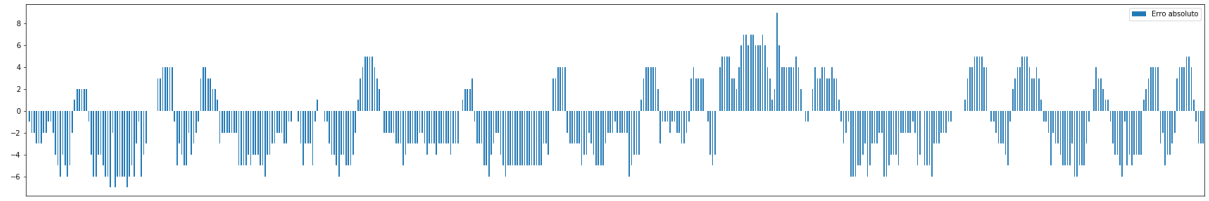

Figura 33 – Erro absoluto da rede LSTM com aprendizado tradicional para cada entrada utilizando o conjunto de teste.

A partir dos gráficos amostrados nas seções 4.2.3.1 e 4.2.3.2, é possível comparar o comportamento dos diferentes modelos. Para os modelos treinados de forma online, é perceptível um atraso nas previsões nas primeiras semanas e um aumento gradual na assertividade do modelo conforme o treinamento vai sendo aprimorado. Este comportamento já era esperado, uma vez que nos primeiros momentos, pouquíssimas amostras são apresentadas ao modelo. Nas Figuras 21 e 28, verifica-se uma redução do erro conforme mais amostras vão sendo treinadas no modelo. Esta diminuição gradual do erro foi maior para o modelo treinado com os dados de *Ceratitis capitata* do que *Anastrepha fraterculus*.

O modelo treinado de forma tradicional para *Anastrepha fraterculus*, tanto na avaliação dos dados do conjunto usado no treinamento quanto do conjunto de testes,

mostrou-se inferior ao modelo treinado online, sendo claramente visível o erro. O modelo fica sempre abaixo do valor real esperado em todos os máximos locais e sempre acima do valor real esperado em todos os mínimos locais, nunca assim alcançando o resultado esperado.

Já para *Ceratitis capitata*, o modelo se comporta de forma mais satisfatória, onde os maiores erros visíveis são encontrados no máximos locais. Nestes casos, o modelo claramente ultrapassa o valor esperado. Analisando o domínio da aplicação, é desejado que, em caso de erro do modelo, o erro seja positivo, ou seja, o modelo informe que há mais insetos do que na realidade há.

Desta forma, pode-se afirmar que o modelo com treinamento tradicional para *Ceratitis capitata* obteve um resultado mais satisfatório que o da *Anastrepha fraterculus* quando seus gráficos de comportamento são analisados visualmente.

Os gráficos informando o erro absoluto de cada modelo mostram que para *Ceratitis capitata*, o modelo com aprendizado online apresenta erros, conforme Figura 21, em uma escala maior variando de -40 à 40. Já os erros do modelo com treinamento tradicional variaram entre -5 à 15, conforme Figura 24. Os erros do modelo online foram mais pontuais, enquanto os erros do modelo tradicional ocorrem em diversas iterações consecutivas. Já para *Anastrepha fraterculus*, o modelo online apresentou, conforme Figura 28, erros mais significativos quando negativos, ou seja, prevendo menos insetos do que a realidade. Os erros variam no intervalo de -30 à 20 e assim como para o outro inseto, são erros mais pontuais, enquanto que o modelo tradicional apresenta erros de -5 à 15, porém com erros consecutivos nas iterações, conforme Figura 31.

Por fim, observando os valores obtidos de MSE demonstrados na Tabela 2, é possível ver que os modelos com aprendizado online obtiveram médias mais baixas que a avaliação realizada sobre os conjuntos de treinamento para o modelo de aprendizado tradicional. Pelo fato da métrica utilizada ser MSE, quanto maior o erro, mais penalizada é a métrica. Logo, obtendo valores menores, os modelos online mostram que, apesar de terem um erro absoluto maior, ele ocorrem menos frequentemente. Por este motivo, apesar de possuírem erros menores, eles ocorrem com maior frequência nos modelos tradicionais, por isso suas médias são maiores para os conjuntos de treinamento. Para os conjuntos de testes, por terem poucas ocorrências de infestações e por corresponderem à 1/3 do tamanho dos dados testados, é esperado que obtivessem um resultado menor para MSE.

5 CONCLUSÃO

Apesar da existência de meios para controle de infestações alternativos menos danosos à saúde dos seres humanos, o uso de pesticidas ainda é o meio mais utilizado. Como o reconhecimento de uma infestação geralmente ocorre em um período onde o número de insetos presentes na lavoura já é considerado alarmante, o uso de pesticidas se torna a melhor alternativa para os agricultores, devido a sua resposta rápida ao problema. Porém, o comportamento dos insetos é previsível. Condições climáticas afetam a forma como os insetos vivem e se reproduzem, tornando assim as alterações climáticas da região uma forma de prever a população de insetos nesta mesma região.

Considerando esta afirmativa, este trabalho propôs o desenvolvimento de um modelo capaz de prever a média de insetos por armadilha presentes na lavoura baseando-se nas alterações climáticas que ocorrem na região. Para o desenvolvimento deste modelo, utilizou-se uma técnica de aprendizagem de máquina com aprendizado supervisionado. A técnica aplicada é conhecida como Rede Neural Recorrente, mais especificamente uma *Long Short Term Memory*, que é uma rede capaz de carregar informação entre suas iterações, sendo assim recomendada para o uso em aplicações onde a sequência temporal dos dados é relevante. Foi utilizada uma base de dados inicial fornecida por pesquisadores do Instituto Biológico de Campinas para a criação de uma base de treinamento mais substancial a partir dos dados meteorológicos fornecidos pela empresa Embrapa.

Este trabalho é a continuação de um trabalho anterior realizado pelo mesmo autor. O presente trabalho apresentou melhorias como: aumento da janela de previsão de 4 dias para 1 semana; inclusão da contagem de insetos presente na lavoura como atributo para a previsão, incluindo assim a informação do crescimento populacional dos insetos; contexto temporal para a previsão, pois o modelo anterior considerava os dados semanais de cada armadilha como independentes entre si o que não reflete a realidade; finalmente, uso de um modelo de regressão, sendo capaz de prever o número de insetos e não mais uma classificação binária indicando se haverá ou não uma infestação. Sendo algumas destas melhorias previstas como trabalhos futuros do trabalho anterior.

Como trabalhos futuros, espera-se ser possível aumentar a janela de previsão para um período de tempo hábil para uma reação onde o uso de pesticidas não seja necessário. Com os resultados atuais, é possível somente reduzir a quantidade de pesticidas aplicados, não havendo tempo suficiente para iniciar a aplicação de um técnica alternativa. Além disso, é necessário também a criação de uma base de dados robusta, com dados que representam o comportamento populacional dos insetos, para avaliação do modelo a partir do treinamento com estes dados.

REFERÊNCIAS

BOLSO, A. de. **Os 10 maiores exportadores agrícolas do mundo**. Accessed: 2019-06-07, https://agriculturadebolso.wordpress.com/2014/12/03/os-10-maiores-exportadores-agricolas-do-mundo/.

CALAMA, R.; FORTIN, M.; PARDOS, M.; MANSO, R. Modelling spatiotemporal dynamics of Pinus pinea cone infestation by Dioryctria mendacella. **Forest Ecology and Management**, [S.l.], v.389, p.136–148, 2017.

CANNON, R. J. The implications of predicted climate change for insect pests in the UK, with emphasis on non-indigenous species. **Global Change Biology**, New Jersey, EUA, v.4, n.7, p.785–796, 1998.

CHOLLET, F. et al. **Keras**. https://keras.io.

CORRÊA-FERREIRA, B. S.; PANIZZI, A. R. **Percevejos da soja e seu manejo**. Londrina/PR, Brasil: Embrapa Soja, 1999.

GERS, F. **Long short-term memory in recurrent neural networks**. 2001. Tese (Doutorado em Ciência da Computação) — Verlag nicht ermittelbar.

HAN, J.; KAMBER, M.; PEI, J. **Data mining: concepts and techniques**. Amsterdam, Holanda: Elsevier, 2011.

HOCHREITER, S.; SCHMIDHUBER, J. Bridging long time lags by weight guessing and "Long Short-Term Memory". **Spatiotemporal models in biological and artificial systems**, [S.l.], v.37, n.65-72, p.11, 1996.

HU, C. C.; LIU, B.; SHEN, J.; LANE, I. Online Incremental Learning for Speaker-Adaptive Language Models. In: INTERSPEECH, 2018. **Anais...** [S.l.: s.n.], 2018. p.3363–3367.

JORDAN, M.; MITCHELL, T. Machine learning: Trends, perspectives, and prospects. **Science**, Washington DC, EUA, v.349, n.6245, p.255–260, 2015.

KHAYAT, C. B. et al. Assessment of DNA damage in Brazilian workers occupationally exposed to pesticides: a study from Central Brazil. **Environmental Science and Pollution Research**, London, Inglaterra, v.20, n.10, p.7334–7340, 2013.

KOCMÁNKOVÁ, E. et al. Impact of climate change on the occurrence and activity of harmful organisms. **Plant Protection Science**, Prague, Republica Tcheca, v.45, n.Special Issue, p.S48–S52, 2009.

LIPTON, Z. C.; BERKOWITZ, J.; ELKAN, C. A critical review of recurrent neural networks for sequence learning. **arXiv preprint arXiv:1506.00019**, [S.l.], 2015.

MAAS, A. et al. Recurrent neural networks for noise reduction in robust ASR. , [S.l.], 2012.

MAIMON, O.; ROKACH, L. **Data mining and knowledge discovery handbook**. London, Inglaterra: Springer, 2005. v.2.

MITCHELL, T. M. **Machine Learning**. 1.ed. New York, EUA: McGraw-Hill, 1997.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. **Sistemas Inteligentes-Fundamentos e Aplicações**, Barueri, Brasil, v.1, n.1, 2003.

MOSTAFALOU, S.; ABDOLLAHI, M. Pesticides and human chronic diseases: evidences, mechanisms, and perspectives. **Toxicology and applied pharmacology**, Amsterdam, Holanda, v.268, n.2, p.157–177, 2013.

MOULY, R.; SHIVANANDA, T.; VERGHESE, A. Prediction models for Bactrocera dorsalis (Hendel)(Diptera: Tephritidae) based on weather parameters in an organic mango orchard. **Journal of Entomology and Zoology Studies**, [S.l.], 2017.

OZAY, M. et al. Machine learning methods for attack detection in the smart grid. **IEEE transactions on neural networks and learning systems**, [S.l.], v.27, n.8, p.1773–1786, 2015.

PORTER, J.; PARRY, M.; CARTER, T. The potential effects of climatic change on agricultural insect pests. **Agricultural and Forest Meteorology**, Amsterdam, Holanda, v.57, n.1, p.221–240, 1991.

RAGA, A.; PAULA, L. Í. S. de; SOUZA-FILHO, M. F. de; CASTRO, J. L. de. Population dynamics and infestation rate of fruit flies in stone fruits in São Paulo State, Brazil. **Annual Research & Review in Biology**, [S.l.], p.1–11, 2017.

RUSSELL, S. J. et al. **Artificial intelligence: a modern approach**. Upper Saddle River, EUA: Prentice Hall, 2003. v.2.

SCHUSTER, M.; PALIWAL, K. K. Bidirectional recurrent neural networks. **IEEE Transactions on Signal Processing**, [S.l.], v.45, n.11, p.2673–2681, 1997.

SOUZA, W. D. de; REMBOSKI, T. B.; AGUIAR, M. S. de; JÚNIOR, P. R. F. A Model for Pest Infestation Prediction in Crops Based on Local Meteorological Monitoring Stations. In: SIXTEENTH MEXICAN INTERNATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE (MICAI), 2017., 2017. **Anais...** [S.l.: s.n.], 2017. p.39–45.

THOMSON, L. J.; MACFADYEN, S.; HOFFMANN, A. A. Predicting the effects of climate change on natural enemies of agricultural pests. **Biological control**, Amsterdam, Holanda, v.52, n.3, p.296–306, 2010.

TROSTLE, R. et al. **Global agricultural supply and demand**: factors contributing to the recent increase in food commodity prices. Washington DC, EUA: US Department of Agriculture, Economic Research Service, 2008.

VENNILA, S. et al. Artificial neural network techniques for predicting severity of Spodoptera litura (Fabricius) on groundnut. **Journal of Environmental Biology**, [S.l.], v.38, n.3, p.449, 2017.

WANG, Z.; BOVIK, A. C. Mean squared error: Love it or leave it? A new look at signal fidelity measures. **IEEE signal processing magazine**, [S.l.], v.26, n.1, p.98–117, 2009.

WILLIAMS, R. J.; ZIPSER, D. A learning algorithm for continually running fully recurrent neural networks. **Neural computation**, [S.l.], v.1, n.2, p.270–280, 1989.

WITTEN, I. H.; FRANK, E. **Data Mining: Practical machine learning tools and techniques**. Amsterdam, Holanda: Morgan Kaufmann, 2005.

ZANATTA, J. F. et al. Interações entre herbicidas e inseticidas na cultura do algodão–Uma revisão. **Revista da FZVA**, Porto Alegre, Brasil, v.14, n.2, 2007.